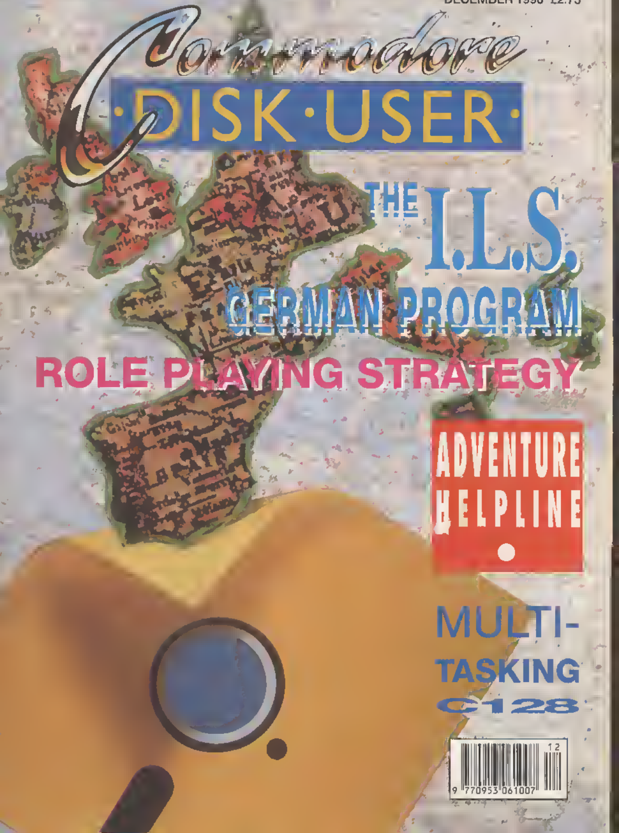
DECEMBER 1990 £2.75

Commodore DISK·USER



THE I.L.S. GERMAN PROGRAM

ROLE PLAYING STRATEGY

ADVENTURE HELPLINE

MULTI-TASKING C128

STAR PRINTERS

	EX VAT	INC VAT
LC10 and LC10-C Superb value printers! 120CPS draft, 30CPS NLQ Multiple fonts	£139	£159.95
LC10 COLOUR The above specification plus full colour	£190	£218.50
LC24-10 24 pin 142 CPS draft. 47 CPS NLQ	£230	£264.50

SPECIAL OFFERS

AMIGA A500 BATMAN PACK OR FLIGHT OF FANTASY PACK

- DELUXE PAINT II
- INTERCEPTOR
- NEW ZEALAND STORY
- BATMAN
- MOUSE
- MODULATOR

PLUS

- 5 EXCITING ORIGINAL GAMES
- JOYSTICK
- 10 3 1/2 INCH DISKS
- DISK BOX

EX VAT 346.75 INC £399

PANASONIC PRINTERS

	EX VAT	INC VAT
KXP-1081 Outstanding value! 120 CPS draft 24 CPS NLQ Friction and tractor feed.	£129	£148.35
KXP-1180 192 CPS draft, 38 CPS NLQ	£160	£184.00
KXP-1124 24 pin 192 CPS draft 38 CPS NLQ	£240	£276.00

SEIKOSHA PRINTERS

	EX VAT	INC VAT
SP-200 190 CPS draft 40 NLQ	£173	£199
SL-92 24 pin 240 CPS draft 80 CPS NLQ	£271.30	£312
SL-1350	£313.00	£360
SL-1350	£433.90	£640
BP-5500	£1169.56	£1345.00

ACE COMPUTERS

MICRO SPECIALIST

BRANCHES THROUGHOUT THE UK

FOR MAIL ORDER CALL 0272 663312

HEAD OFFICE
30 Cannon Street, Bedminster,
BRISTOL BS3 1NN.
Tel: 0272 663312
Fax: 0272 231466

HOME COMPUTERS
AND EQUIPMENT REPAIRS
3 North Street, Bedminster,
BRISTOL.
Tel: [illegible]

87 City Road,
CARDIFF.
Tel: 0222 463060/471600

54 Commercial Road,
NEWPORT (Gwent).
Tel: 0633 212176

16 West Street,
WESTON-SUPER-MARE.
Tel: 0934 419040

31 Farringdon Road,
SWINDON.
Tel: 0793 512073/4

36 Robertson Street,
HASTINGS.
Tel: 0424 422929

92 Upper Parliament Street
NOTTINGHAM
Tel: 0602 473873

3/5 Sambourne Road
WARMINSTER
Tel: 0985 846131

ALL PRICES are subject to alteration without notice and **INCLUDE FREE NEXT DAY DELIVERY** within mainland Great Britain

ACE SERVICES TO BUSINESS

Ace Computers are able to advise their business customers over the whole range of computer needs. Our services include analysing the problem and recommending computer solutions; supplying computer systems including all equipment and programmes, installation of equipment and training.

We offer a range of financing facilities – written details on request – and our maintenance contracts guarantee next day service throughout the country.

Whether you are computerising for the first time, or updating existing systems, ACE will be most pleased to hear from you!

Call our Customer Liaison Desk at the nearest shop, or on our Bristol number

SPECIAL MESSAGE TO RETAILERS

ACE are able to supply EPOS and computerised stock control systems to suit from one to eight shops. These systems can revolutionise your work load and substantially reduce the costs of financing stock.

Our comprehensive service includes programme modification to meet your needs; supply and installation of equipment and training.

Financing facilities are available – written details on request – and our maintenance contracts guarantee next day service.

Prices will surprise you! Call ournearest shop or our Bristol number and ask for the Customer Liaison Desk for more details.

FRANCHISE SERVICES

Ace Computers have a highly successful franchise operation. If you are interested in opening your own comuter shop, please call us and ask for a copy of the Franchise Brochure.

WE HAVE A NUMBER OF SHOPS READY TO START IN BRISTOL AND NOTTINGHAM

PHILIPS MONITORS

	EX VAT	INC VAT
7502 Mono Leads included	£76	£87.76
8833 Colour Leads inc. ST & Amiga compatible	£225	£255.75

3 1/2" HIGH QUALITY DS/DD DISKS INC. FREE BOX FOR £9.50

Commodore DISK·USER

Volume 4 Number 2 DECEMBER 1990

IN THE MAGAZINE

ON THE DISK

Publisher: Hasnain Walji
Group Editor: Paul Eves
Technical Assistant: Jason Finch
Publishing Consultant: Paul Crowder
Advertisement Manager: Cass Gilroy
Designer: Mark Newlon
Distribution: Seymour Press Distribution Ltd. Winsor Honse, 1270 London Road, Norbury, London SW16 4DH. Tel:081 679 1899. Fax: 081 679 8907
Printed By: Gibbons Barford Print

Subscription Rates

UK	£33.00
Europe	£39.00
Middle East	£39.30
Far East	£41.60
Rest of World	£39.70 or $69.00

Airmail rates on request
Contact: Select Subscriptions. Tel: (0442) 876661

Commodore Disk User is a monthly magazine published on the 3rd Friday of every month. Alphavite Publications Limited, 20, Potters Lane, Kiln Farm, Milton Keynes, MK11 3HF Telephone (0908) 569819 FAX. (0908) 260229 For advertising ring (0908) 569819

EDITORS COMMENT

I THINK THAT THE HARDEST PART OF MY JOB IS KEEPING EVERYONE HAPPY AND INTERESTED IN THE MAGAZINE. TAKE THIS MONTHS ISSUE FOR INSTANCE. YOU WILL NOTICE THAT THERE ARE NOT THE USUAL 10 PROGRAMS ON THE DISK. IN FACT, STRICTLY SPEAKING, THERE ARE ONLY 3 PROPER PROGRAMS, AND TWO OF THESE ARE FOR THE C128. HOWEVER, THE READING MATERIAL IS BOTH INFORMATIVE AND INTERESTING, (AND EVEN HUMOROUS!!).

TWO THINGS THAT ARE BECOMING APPARANT, JUDGING BY THE SURVEYS THAT WE HAVE HAD RETURNED SO FAR, IS THAT A LOT OF YOU THINK THAT THE MAGAZINE IS TOO SERIOUS, AND DOESN'T CATER FOR THE NOVICE. REMEMBER THIS, WE CAN ONLY PUBLISH WHAT WE RECEIVE. SO IF YOU THINK THE PROGRAMS ARE TOO DIFFICULT TO UNDERSTAND, THEN IT IS UP TO YOURSELVES TO MAKE THEM MORE EASILY UNDERSTANDABLE, FOR BOTH THE EXPERIENCED AND THE NOVICE.

THE OTHER POINT IS THE LACK OF MAIL ORDER ADVERTISEMENTS. AGAIN, WE CANNOT FORCE PEOPLE TO ADVERTISE WITH US. IF THEY PREFER OTHER MAGAZINES THERE IS NOT MUCH WE CAN DO ABOUT IT. ANYWAY, I HOPE YOU STILL ENJOY THIS MONTHS ISSUE

DISK INSTRUCTIONS

Although we do everything possible to ensure that CDU is compatible with all C64 and C128 computers, one point we must make clear is this. The use of 'Fast Loaders', 'Cartridges' or alternative operating systems such as 'Dolphin DOS', may not guarantee that your disk will function properly. If you experience problems and you have one of the above, then we suggest you disable them and use the computer under normal, standard conditions. Getting the programs up and running should not present you with any difficulties, simply put your disk in the drive and enter the command.

LOAD"MENU",8,1

Once the disk menu has loaded you will be able to start any of the programs simply be selecting the desired one from the list. It is possible for some programs to alter the computers memory so that you will not be able to LOAD programs from the menu correctly until you reset the machine. We therefore suggest that you turn your computer off and then on again, before loading each program.

HOW TO COPY CDU FILES

You are welcome to make as many of your own copies of CDU programs as you want, as long as you do not pass them on to other people, or worse, sell them for profit. For people who want to make legitimate copies, we have provided a very simple machine code file copier. To use it, simply select the item FILE COPIER from the main menu. Instructions are presented on screen.

If for any reason the disk with your copy of CDU will not work on your system then please carefully re-read the operating instructions in the magazine. If you still experience problems then:

1. If you are a subscriber, return it to:
Select Subscriptions Ltd
5, River Park Estate
Berkhamsted
Herts
HP4 1HL
Telephone; 0442 876661

2. If you bought it from a newsagents, then return it to:
CDU Replacements
Interceptor Group
Mercury House
Calleva Park
Aldermaston
Berks
RG7 4QW
Telephone; 0734 817421

Within eight weeks of publication date disks are replaced free.

After eight weeks a replacement disk can be supplied from INTERCEPTOR GROUP for a service charge of £1.00. Return the faulty disk with a cheque or postal order made out to INTERCEPTOR GROUP and clearly state the issue of CDU that you require. No documentation will be supplied

Plese use appropriate packaging, cardboard stiffener at least, when returning disk. Do not send back your magazine, only the disk please

NOTE: Do not send your disks back to the abov address if its a program that does not appear t work. Only if the DISK is faulty. Program fault should be sent to: BUG FINDERS, CDU, Alphavite Publications Ltd, Unit 20, Potters Lane, Kiln Farm Milton Keynes, MK11 3HF. Thank you.

I.L.S. (THE GERMAN PROGRAM)

With 1992 fast approaching!!!

JOHN BASKERVILLE

wouldn't it be nice if we could speak at least one other language? This program, with the aid of your C128, helps you on your way

For most people, learning to speak a foreign language is not the easiest thing in the world to do. There are always little things that never seem to fit into place, or odd words whose meanings always slip the mind. As with most things, perseverence is the key.

THE STRUCTURE OF LANGUAGE

When learning a language you have to come to grips with both grammar and vocabulary. The grammatical rules of a language dictate how its sentences are put together, how tenses are used, when capital letters begin words and many other things. Someone who had learnt a large foreign vocabulary would arguably get nowhere without an understanding of how to use it. On the other hand, someone who knows every grammatical rule in the book and is probably more technically accurate at a language than 90% of its native speakers, would get nowhere without a good solid vocabulary to work with. So, we have two major areas to learn about, each almost useless without the other. But what is the best way to learn? This is where I.L.S. may be able to help you, not with grammar, but with vocabulary - often the biggest problem.

INTERACTIVE LANGUAGE SYSTEM

I.L.S. stands for Interactive Language System. When I started to write I.L.S. I wanted to design a program to help me learn vocabulary. I was being taught GERMAN at school but was having problems learning the words. I had in mind a program that would both teach and test, so as to provide a way of learning and of assessing that learning. I.L.S. will teach you, test you and then re-teach those words you have not yet learnt. Thus the system interacts with you, teaching to your needs.

USING THE PROGRAM

The instructions on how to use this program follow in four parts;

SECTION 1 - Introduces you to a few general points about the program. It shows you how to move around, how to enter data etc.

SECTION 2 - Tells you how you can create and edit your own files.

SECTION 3 - Shows you what you can do with files once they have been created. Introduces you to the LEARN and TEST modes and to the word search facility.

SECTION 4 - Details the program's disk commands and file information facilities

PLEASE REMEMBER - The program is very 'user friendly', as long as you read all on-screen information and think before you act, it will be hard for you to do anything wrong.

SECTION 1 - GENERAL SECTION

I.L.S is a very user friendly package and relies heavily on a menu system for access to its many functions. Using the menus is extremely simple. At the prompt - Press appropriate key to continue - you press the key corresponding to the option you wish to select. Nothing could be easier. The 'EXIT' option exists on most menus and will usually return you to the previous menu or the main menu. (The one you see upon loading the program)

INPUTTING INFORMATION

Practically everything you do with 'THE GERMAN PROGRAM' will require some form of text input;

filenames, disk titles and not forgetting the words themselves. Any time the program needs a textual input you will see a prompt and the input line appear on the screen like this:-

```
Please enter the filename
_________________________
?????          ?????
```

At this point you enter the text as normal and press return when you have finished. Whilst entering text, all the usual editing facilities are available. You should already know the most common of these; delete, insert and the cursor keys. Listed below are some of the less known, but equally useful functions.

HOME - Moves the cursor to the beginning of the line.
CLR - Clear entire input box.
ESC A - Enable auto insert mode.
ESC C - Disable auto insert mode.
ESC K - Move cursor to end of current line.
ESC P - Delete to start of line.
ESC Q - Delete to end of line

When inputting a phrase, the input box consists of two lines rather than one. Some facilities are useful only in this environment.

HOME - Moves cursor to top left corner of input box.
ESC J - Moves cursor to start of current line

SPECIAL CHARACTERS

Some German words contain special characters such as UMLAUTS (two dots that appear over certain vowels). All the characters you may need are catered for by the program. Below is a list of the keys to press to get certain characters.

UMLAUT

LETTER	CORRESPONDING KEY
'a'	CBM and 'a'
'o'	CBM and 'o'
'u'	CBM and 'u'
'A'	Press '@'
'O'	Press '*'
'U'	Press '^' (up arrow)

You can get an S-set (symbol for double s) by pressing CBM and 's'. Please note that the program will not recognise a double 's' and an S-set as the same thing in a test situation. You must always use the same letters in a test as you did to enter the word in the first place. If you are going to use the S-set then use it properly or not at all. Another symbol I have created is a dash with an UMLAUT on it. This is often used when entering plurals. This character can be obtained by pressing CBM and '-'

GENERAL QUESTIONS

Sometimes the program may ask you a question such as 'Input more data?'. All these questions require a simple 'yes' or 'no' answer entered merely by pressing 'Y' or 'N' All of these questions have a default answer (that is to say that if the default is 'yes' then pressing any key other than 'N' will result in a 'Yes' decision). The default answer varies depending on the question, details of the default answers are given at the relevant points later in these instructions.

FILE FORMAT

Each file used with I.L.S. is made up of four sections, each of which is made up of a number of records The four sections which make up each file are VERBS, NOUNS, DESCRIPTORS (adjectives, adverbs etc) and PHRASES. It is necessary to have these different sections because each data type has a different format. Each record in the VERBS section is made up of five entries, these are as follows,

ENGLISH - The English meaning of the VERB.
INFINITIVE - The German infinitive of the VERB.
3rd PERSON - 3rd person singular form.
PERFECT - Perfect tense form.
IMPERFECT - Imperfect tense form.

Each record in the NOUNS section is made up of four entries, these are as follows;

ENGLISH- The English meaning of the NOUN
DEFINATE ART -The appropriate definate article for the NOUN.
NOUN - The German version of the NOUN.
PLURAL- The plural form.

In any language, as well as VERBS and NOUNS there are many types of describing words, ADJECTIVES and ADVERBS for example. To cater for these words, I.L.S. has a file section called DESCRIPTORS. Each record in this section is made up of two entries. These are simply the English and the German. The same format works for the PHRASE section, the only difference being that PHRASES can be up to two lines long.

LEARN FILE

When you are using the LEARN MODE, the computer looks up each record to see if it is included in the 'LEARN FILE'. When you load a file, all of its records are included in the learn file. At some point you may wish to exclude some words from the learn file. This can be done easily using the 'VIEW and EDIT FILE' mode (see

'EDITING YOUR OWN FILE')

SECTION 2 - CREATING AND EDITING USER FILES

CREATING YOUR OWN FILES

To set about creating your own file from scratch could not be easier. First, make sure you have a formatted disk ready to save the new file when you have finished. Now select 'CREATE YOUR OWN FILE' from the main menu by pressing 'A'. At this point you will be presented with four choices; VERBS, NOUNS, DESCRIPTORS and PHRASES. Select which data type you wish to enter first by pressing the appropriate key. From this point you proceed as follows;

1. Follow the on screen prompts and enter the correct words at the appropriate time. The computer is very helpful here. It will tell you 'Input the English' or 'Next type in the German' or whatever is appropriate.
2. When you have entered one record the computer will ask you 'Do you wish to make alterations?' (default answer is 'no'). If you select 'Y' you will be asked to re-enter the record.
3. The computer now asks you 'Input another record?' (default answer is 'Y'). If you select 'N' you will be returned to the create mode menu.

When you wish to exit the create mode, select 'EXIT' from the menu. You will now be asked to input a filename so that the computer can save you new file Enter a suitable filename, then follow the on screen instructions and your file will be stored on disk

EDITING YOUR OWN FILES

Once you have created your own files you may find that you need to edit them, perhaps to correct a mistake. To edit a file you must first load it, then select 'View and Edit file' from the main menu by pressing 'B'. You will nowbe presented with the now familiar menu asking whether you wish to look at VERBS, NOUNS, DESCRIPTORS or PHRASES. Make your choice and the first record in that section will appear on the screen. At this point you have a number of options;

1. Using the cursor keys you can choose to look at a different record. 'UP' and 'DOWN' change the record number in jumps of 10. 'LEET' and 'RIGHT' change the record number by steps of 1.
2. Pressing 'D' will delete the current record and move the last record in the section down to fill the gap.
3. Pressing 'E' will let you edit the current record. Just follow the on screen instructions.
4. At the bottom of the screen is a line which tells you whether or not the record is included in the learn file. By pressing 'L' you can toggle this between 'YES' and 'NO'
5. Pressing 'RETURN' will take you back to the main edit mode menu.

When you exit 'View and Edit mode' If you have made any alterations to the file, the computer will ask you to insert the disk which contains that file and press a key. At this point the computer will save the alterations, thus making these permanent, before returning you to the main menu.

EXTENDING A FILE

If at some point you wish to add some words to an already existing file you will need to use the EXTEND FILE facility. To use this mode you must have a file resident in memory. At this point the extend file mode can be accessed by pressing 'C' on the main menu. You will be presented with the VERBS, NOUNS, DESCRIPTORS and PHRASES menu. Having chosen which section you wish to extend you continue by following the on screen prompts and entering the words in the usual way. (see 'CREATING YOUR OWN FILES' if you are unsure).
When you select 'EXIT' from the extend mode menu, the computer will ask you to insert the disk which contains the current file. When you have inserted the disk, press a key and the computer will save the file extensions.

SECTION 3 - USING EXISTING FILES

LOAD FILE

To load a file select option 'E' from the main menu. 'The German Program' will warn you that you will wipe out any file still in memory and give you the chance to abort to the main menu. If you choose to go on you will be asked to enter the filename of the file you wish to load. You then insert the LOAD disk that contains that file, press any key and wait for it to load.

WORD SEARCH

The word search is a useful facility which can be used as a dictionary to check up on the meaning of various words. The word search facility is selected from the main menu by pressing 'M'. The program then provides you with the search menu, to carry on from here you just follow these instructions;

1. Select whether you wish to search for a VERB, NOUN, DESCRIPTOR or PHRASE. If you are not sure what category your word falls into then simply select 'UNKNOWN' from the menu.
2. Select whether you wish to input the German or the

English.
3. Enter the word or phrase you wish to search for.
4. Enter the filename of the file you wish to search.
5. At the prompt, insert the appropriate disk and press any key.

The program will then search through the selected file for your word. If a match is found then the computer will display all the relevent information on the screen. If no match is made then the computer will give you the option to search another file. (default answer is 'no'). If you choose to do so then you merely repeat the above process starting at instruction 4.

N.B: Using the word search has no affect on any file in the memory. You could have a file in memory called 'School' and search another file called 'Home' without damaging 'School'.

LEARN MODE

The whole idea of I.L.S. is to help you learn vocabulary, so this mode, along with the test mode, is the most important of the program's facilities. I.L.S. uses a very simple learning technique called 'READ and REMEMBER'. The program displays the German word in the centre of the screen for a short period of time. This is then replaced with the English version of the same word, again for a short period of time. This process is then repeated one more time. As the words appear on the screen you should say them out loud to yourself. It's no use trying to learn words this way if you do not give 100% concentration. 'Read and Remember' will not work if you are listening to the radio or watching T.V. at the same time. It may seem a strange way to learn words, the speaking to yourself bit may seem silly, but I guarentee that if used properly it really does work. Having used the program myself I would suggest only learning a few words at a time, probably less than fifty, otherwise your concentration may wonder and you will not learn anything.

Before you can use learn mode you must have a file in memory (see LOAD FILE). You then select it from the menu by pressing 'E'. On the screen will appear the LEARN MODE MENU, from here you proceed as follows,

1. Select which section of the file you wish to learn (V,N,D or P).
2. Sit back and get ready to concentrate.
3. When you are ready, follow the on screen command and press a key.
4. As the words appear on the screen, read them aloud.
5. When the program has gone through all the words, follow the on screen command to return to the menu.

When learning some words, they appear on the screen in the following order, depending on whether you are learning VERBS, NOUNS, DESCRIPTORS or PHRASES;

VERBS- ENGLISH - INFINITIVE - IRREGULAR - PERFECT - IMPERFECT
NOUNS - ENGLISH - GERMAN - PLURAL
DESCRIPTORS - ENGLISH - GERMAN
PHRASES- ENGLISH - GERMAN

N.B. The computer only displays words which are included in the learn file (see general information - learn file)

When learning VERBS, if the computer comes across an entry containg a dash ('-') then it will miss it out. For example if the the record was;

ENGLISH : To help
INFINITIVE: Helfen
IRREGULAR : Hilft
PERFECT : -
IMPERFECT : -

The computer would display;
'to help; - 'helfen' - 'hilft' and then go back to 'to help'.

TEST MODE

Another important part of the learning process is the test mode as it gives you some idea of how well you are doing. (Or not doing, as the case may be). Test mode is selected from the main menu by pressing 'G'. This takes you to the Test Mode Menu - from here you proceed as follows;

1. Select what category you wish to be tested on. (V,N,D or P)

2. Select whether you wish to have an English - German test. (The computer provides the English, you the German) or a German - English. (vice-versa)
3. Select which type of test you wish to have.
a) Test on all - just what it says, a test on everything.
b) Test on selected - tests only on the words included in the learn file.
c) Random test - here you choose a number using the cursor keys. (as previously). You are then given a test with that number of questions.
4. Take the test! Just follow the on screen commands and try your best.

When the test is over the computer will display your score in terms of a percentage. After pressing a key you will then be given the option to 'Change Learn File' (default answer is 'no'). If you answer 'yes' then the computer goes through the learn file, unselecting all of the words that you got right.

N.B. When doing a VERB or NOUN test you can miss sections of the word out by pressing return. For example, if you are doing a NOUN test but are not concerned about the PLURAL form, on each question when asked to input the PLURAL form, just hit return. This will result in the computer ignoring the PLURAL for that question.

SUGGESTED LEARNING METHOD

For this example please Imagine you wish to learn all the NOUNS in a file called 'FOOD'.

1. Use 'Load File' to load 'FOOD'.
2. Use 'Learn Mode', once, to learn the NOUNS.
3. Go to Test Mode and do a test on all.
4. Change the learn file.
5. Use learn mode to relearn the words you got wrong.
6. Go to test mode and do a test on selected.
7. Return to step 4 until you get a score of 100%
8. Do a random test on at least 75% of the NOUNS - you should get a score of over 90%. If you don't then return to step 1 (loading the file again is the quickest way to set the entire learn file to 'yes').

For even a smallish file, this proecess could take quite a while - but if you do it properly it will be worth it.

SECTION 4 - EXTRAS

FILE INFORMATION

If you want to know how large the file you are currently working on is, then just select option 'D' (current file info) from the main menu. This will tell you how many VERBS, NOUNS, DESCRIPTORS and PHRASES you have entered. It will also show how much memory you have left.

DISK COMMANDS

These are selected from the main menu by pressing 'I'. I.L.S. offers most of the disk commands you might need whilst using the program:

1. DIRECTORY - Allows you to view any disk. Simply follow the on screen instructions. Please note that I.L.S. files save as four seperate parts. Each part is ended with an identification letter consisting of v,n,d or p.

2. FORMAT Will format a disk in the usual way. You will be asked to supply a disk title but the program will use the letters GP (German Program) as the disk ID.

3. SAVE - This option is useful for making backups of your files. It allows you to save the file currently in memory. You will be asked whether you wish to use the existing filename or give the file a new one.

4. RENAME - Allows you to rename a file. You will be asked to enter the old filename and the new one. (please note that this only works for I.L.S. files).

5. SCRATCH - Will delete a file from the disk. You have to supply the filename. Again, this only works on I.L.S. files.

6. COLLECT. This will collect any wasted space on your disk. Please refer to your disk manual for more information on the uses and dangers of this command.

DISK ERRORS

Should a disk error ever occur whilst the program is operating, I.L.S. will inform you of what the problem is and then return to the main menu. IT WILL NOT TRY TO RESUME THE OPERATION THAT FAILED. If an error occurs at any time whilst the program is saving your file you can still retrieve your data by correcting the problem and then using the SAVE option from the Disk Commands menu.

ON THE CDU DISK

On the CDU disk there are four files for I.L.S. They are;

1. LOADER - This program loads and runs the main I.L.S. program.

2. CHARGEN - This program creates the character set that the I.L.S. program uses to get special German characters such as UMLAUTS.

3. I.L.S - The main program.

4. GCHAR - The redifined characters produced from CHARGEN.

To get the program up and running simply insert your disk and enter the command; RUN"LOADER". I hope you enjoy using this program as much as I have...

FURTHER ADVENTURE'S IN "C"

POINTERS, ADDRESSES AND ARRAYS.

JOHN SIMPSON

The 'C' language series is well underway. We look at Pointers, Addresses and Arrays in this months installment

Before we go on any further, I must put the record straight. It has come to our attention that when we started this series, back in June, the very first four lessons were unfortunately missed out. Therefore, reproduced for you now are those missing FOUR lessons.

The first program we shall construct in the C language can be considered as one of the smallest of programs It doesn't actually do anything except to serve as a demonstration of one or two points.

```
/* lesson 1 - a very small C program

main()
{
}
```

The program starts with a comment line. In Basic when we wish to make a comment we would use the REMark statement such as;

```
10 REM lesson 1
```

The C equivalent of REM uses the symbols /* to start the comment line and terminates the comment with */

Because C does not use line numbers, I will be using comment lines quite profusely throughout this series An important note is that comment lines can be used enywhere within a program.

Follo ing the comment ine we have a single FUNCTION which is named, in this case, MAIN(). The point of the brackets will be explained later. I could have named the FUNCTION last(), first(), adam(), eve() or filecreate(). I chose main() because it's the only aunction in this short program and a MAIN() function MUST appear somewhere within a program, (ANY 'C' program), for the program to compile correctly. It does not have to appear at the begining, just so long as it appears somewhere.

Next came a pair of braces {}. These are used to enclose the lines of code for the function block. So far, I haven't placed any lines of code in the program. I could have placed the braces immediately after MAIN(){}. It makes no difference, but for the sake of clearness, readability and set standards, it is conventional to follow a neat layout

```
/* Lesson 2 - printing to the screen
in C */

main()
{
 printf("hello everyone\n")
}
```

Lesson 2 starts as before with our comment line, the main() function, and the pair of braces {} However, now we have placcved a line of code between the braces Here we have used the command PRINTF() This is the standard output command to print to the screen Within the brackets any characters to be printed on the scren are placed between the quotes.

"hello everyone"

C prints in a stream fashion, one character after another, and does not automatically terminate a line of code with a carriage return or line feed For example, if between the braces we had placed:

```
printf("hello"),
printf("every"),
printf("one");
```

The resulting printout would appear as; helloeveryone.

The symbol \n is the standard instruction to tell the compiler when to generate a line feed.

The line then terminates with a semi-colon (;) Every separate C command must always terminate with a semi-colon. After using a colon () to terminate in Basic, and a semi-colon to instruct - do not execute a line feed - this may take a little memory effort. Remember, in Basic you use a colon, whereas in C you use a semi-colon. Finally we close the function with the terminating brace.

```
/* Lesson 3 - something about
tabulation */

main()
{
  printf(\tThis is tabulation at
work\n");
}
```

Here everything is the same as lesson 2 other than the text change, that is, I have introduced a \t at the start of the text. This will tabulate the cursor position on the screen. It will tabulate each eighth character position, similar to the use of the comma in Basic. \t\t would tabulate to the sixteenth position

```
/* Lesson 4 - Some variable output */

main()
{
  int content;  /* command line 1 */
   content=50;    /* command line 2

      printf("The street contains %d
houses.\n",content);  /* command line
3 */
}
```

Lesson 4 starts off with our usual comment line, and the only function being main(). However, now I've introduced three command lines, each terminated with a semi-colon.

Line 1 - int content; This shows we have set a variable called conent to be of an integer calue, int.

Line 2 - Content=50, This sets the variable content, to the value of 50

Line 3 - Our print instructions will print out the message and will add the value represented by the variable, content, which follows the comma after closing the text within quotes.

The symbol %d indicatesthat it is of an integer quantity and it's position within the text where it will be printed. We can also output multiple integers, for example:

```
printf("There are %d houses in Street
%d and %d houses in Street
%d\n",content1,street1,content2,street2
```

This would print;

There are 50 houses in Street1 and 60 houses in Street 2. So long as we had placed 50 in content1, 1 in street1, 60 in content2 and 2 in street2.

Back to Line 2 where I set the integer value of 50 to the variable called content. A VERY important point which must be remembered and observed is the ASSIGNMENT OPERATOR =. This is NOT the same as Basic. Here it assigns a value to a variable and must not be confused with the EQUALITY OPERATOR ==, which IS the same as the Basic =

We apologise for missing out these important 4 lessons. I should imagine most of you will have been wondering what the heck has been going out this last couple of lessons. Anyway, back to this months installment. . .

POINTERS AND ADDRESS

As I have said, somewhere earlier, a pointer is a variable which contains the address of another variable or constant (I shall refer to this as an object), Sometimes the object pointed to will be an object within an array, and since the pointer contains the address of the object, it becomes possible to indirectly address the object from the pointer

Let us say that OOF is an integer variable, and that POOF is a pointer, then the operator '&' will give the address of an object.

```
POOF = &OOF;
```

Here we have assigned the address of OOF to the variable POOF, and so POOF now points to OOF

The unary '&' operator can only be used with variables and array elements. It is illegal to use constructions such as &(OOF+1), &2, as is also the taking of the address of a register variable

Next we have the unary '*' operator. This takes the address of the target, and will access that address to fetch it's content. Therefore, if DOOD is also an integer variable, then DOOD=*POOF; will assign to DOOD the content of whatever POOF is presently pointing to.

So

```
POOF = &OOF;
DOOD = *POOF;
```

will assign the same value to DOOD as will DOOD = OOF;

We must also declare all the variables:

```
int OOF, DOOD;
int *POOF;
```

Note that a pointer is constrained to point to a particular kind of object

```
ie.int *POOF,
  double *POOF;
  char *POOF
```

A pointer is allowed to occur within all expressions. As an example of this, if POOF points to the integer OOF then *POOF is allowed to appear in any context where OOF might:

DOOD=*POOF+1; sets DOOD to one more than OOF.

printf("%DOOD",*POOF), prints the current value of OOF

DOOD=sqrt((double)*POOF produces the sq. root of OOF

If POOF points to OOF then, *POOF=0 sets OOF=to

ero, and *POOF +=1 or (*POOF)++ will increment OOF. We require the use of parenthesis otherwise the expression would increment POOF instead of what it points to.

Because pointers are numbers we can manipulate them as we can any other number. For example: rloof = POOF; copies the content of POOF into rloof and so forces rloof to point to whatever POOF is pointing to.

POINTERS AND ARRAYS

Because the relationship between pointers and arrays is so strong, they should really be treated simultaneously!

Let us declare an array

int ages[10] /* note the use of square brackets! */

This declaration defines an array called ages to the si

e of ten objects, or elements. In other words a block, or row of objects, ten deep.

ages[0], ages[ ], ages[2]
ages[9]

The first element, or object of an array is ALWAYS located at element
ero.

If we then declare a pointer, int *ptr~ages; and assign ptr ages = &ages[0] we have set ptr_ages to point to the first, or
eroth, element of ages. In other words ptr_ages now contains the address of ages[0].

The assignment, x=*ptr_ages; will now copy the content of ages[0] into x.

If we write (ptr_ages+1); we are retenng to the content of ages[1], and ptr_ages+1 is the address of ages[1]

If a particular element of an array is pointed to by ptr_ages then ptr_ages+1 points to the next element, and ptr_ages-1 points to the preceeding element

ptr ages=&ages[0] can also be written as ptr_ages=ages.

A reference to ages[x], can also be written as *(ages+x). When evaluating ages[x], the compiler will automatically convert it to *(ages+x) The two forms are equivalent. Applying the operator '&' it follows that &ages[x] and -ages+x are also identical; ages+x is the address of the x_th element beyond ages.

When an array name is passed to a function, that which Is passed is the location of the beginning of the array. Once within the called function this argument becomes a variable like any other, and so an array name argument is truly a pointer, or, a variable containing an address

Check this small function which follows and will evaluate the length of a string:

```
stringlen(alpha) /* returns the length of a string */
char *alpha,
|
   int lngth;
   or (lngth = 0; *alpha != '\0'; alpha++)
      lngth++;
      return(lngth),
   |
```

We can quite legally increment alpha because it is a pointer variable - alpha++ has no effect upon the character string within the function which called stringlen It merely increments stringlen's personal copy of the address.

char *alpha and char alpha[] are equivalent, which of the two is used is determined mainly by how the expressions are written within a function.

It is also quite legal to pass only part of an array to a function by passing a pointer to the beginning of the 'sub-array'.

eg.adam is an array, so:

stringlen(&adam[2]) or stringlen(adam+2)

pass to the stringlen function the address of the third element of the array adam (remember all arrays start at element
ero)

Within the function stringlen, the argument declaration could read:

```
stringlen(array)   or   stringlen(array)
int array[];            int *array;
{                 ..  |
 . .              ..  .....
[                 ..  ]
```

So far as the function is concerned it is of no consequence that the argument is only a part of a much larger array.

CHARACTER POINTERS AND FUNCTIONS

A string constant written as

"I am a string"

is an array of characters. I wrote of this somewhere earlier, where I explained that the compiler will always terminate the string with a null character, ('\0'). The length of the character array is, then, one element longer than the string itself.

Probably the most often occurence of string constants are within the arguments to such functions as:

printf("Well, this is really i , then.\n");

Access to a character string which appears within a program, such as the foregoing, is made via a character pointer, and this is exactly what the function printf() receives - a pointer to a character array where the string is contained.

Of course character arrays need not be functions arguments as the foregoing example was, but can be declared:

```
char *message;
   message = "Well, this is really it, then.";
```

Here a pointer to the actual characters has been assigned to message. This is not a copy of a string; only pointers are involved. C does not provide us with any operators for processing an entire string of characters as do many other languages, such as Basic. This might, at first sight, appear as a drawback but, as we delve ever deeper into the intricacies of C we will discover otherwise.

Let us now study a very useful function from the standard I/O library.

```
strcpy(source,target) /* this function copies the target string */
                      /* to the source string */

char *source, *target;
{
   while ((*source++ = *target++) != '\0');
}
```

Because arguments are passed by values the function strcpy can use SOURCE and TARGET in any way it pleases.

Here the value of *target++ is the character in the string which TARGET pointed to before TARGET is incremented (remember, postfix ++ doesn't change TARGET until after the character has been fetched). In the same way the fetched character is stored into the SOURCE position before SOURCE is incremented. The loop is controlled by testing the character against \0 (string terminator). The effect of this is that each character in turn is copied from TARGET to SOURCE up to and including the terminator, \0

Finally we can observe that a comparison with \0 is redundant, and so the function is often written without the \0 test. Although this may seem odd at first glance, a little thought will produce the obvious, and the notational difference is considerable. The idiom should be mastered, if for no other reason than it is frequently observed in C programs.

So, the while loop can be changed to:

```
while(*source++ = *target++);
```

MULTI-DIMENSIONAL ARRAYS

C does provide for a multi-dimensional array system (arrays of arrays), although in practise it is often that an ARRAY OF POINTERS is more widely used (I shall come to this shortly).

Let us examine a small program using a two dimensional array which simply lists some columns of items. We will fill and dimension the array in one operation.

First, we need to declare the array. This is done thus:

```
static int list[5][4] = {
```

Note the subscripts of the array are each placed within seperate square brackets, is followed by an equal operator, and an opening brace. It is not, as one might suspect from other languages, like this:

```
static int list[5,4] = { /* incorrect */=
```

Note also that the array is declared STATIC

Now we can add each row of array elements, which follow the opening declaration

```
{20,40,82,13},
{81 73,23,22},
{67,30,72,89},
{89,37,81,73},
{23,33,94,18}
};
```

Or if we were dealing with a two dimensional array showing two longer sets of items, such as the days in the month of a normal year, and a leap year, we would declare our array:

```
static int day_list [2][12]={

{31,28,31,30,31,30,31,31,30,31,30,31
},

{31,29,31,30,31,30,31,31,30,31,30,31
}
};
```

Note that each array row is contained within braces, that each row, apart from the last, is ended with a comma, and finally a closing brace to match the opening brace after the equals sign of the array declaration, followed by our usual semi-colon.

Right... now for the entire program

```
main()
{
static int list[5][4]={
  {20,40,82,13},
  {81,73,23,22},
  {67,30,72,89},
  {89,37,81,73},
  {23,33,94,18}
  };

int x,y

  printf("Item. Beans. Peas Spuds
Carrots\n\n");
      for (x=0, x<=3; x++)
    printf("%d",x+1); /* item number
*/
      for (y=0; y<=4· y++)
    {
      printf("%d",list[x][y]); /* items
in four columns */
    }
      printf("\n");
    }
  }
```

The use of braces and the requirements for using a static variable/constant is confined to arrays.

After declaring and initialising the array two more integer variables for use as counters, x/y, are declared for the array print out. Probably the only real headache comes when actually arranging the screen format.. we will deal with formatting later in the series.

If we need to pass a two dimensional array to a function then we must include the column dimension, the row dimension is irrelevent since what is passed is a pointer. In the example program it is a pointer to objects which are arrays of four int's. Therefore, if the array LIST is to be passed to a function called do_list the declaration of do_list would be.

```
do_list(list)
int list[5][4];
{
  .....
}
```

The argument declaration in the function do_list could be:

```
int list[][4];
```

since the number of rows is irrelevent, or it could be:

```
int (*list)[4];
```

which declares the argument is a pointer to an array of four integers. We must use the parenthesis, since the [] square brackets have a higher precedence than * does. Without the parenthesis, the declaration

```
   1[4];
```

s an array of fo pointers to integers, which neatly takes us to:

POINTERS ARRAYS; POINTERS TO POINTERS

Because pointers are themselves numbers, it naturally follows that we can therefore set up an array of pointers.

It must be said that it is more simple to use an array of integers, and they would take less space than an array of pointers, however, arrays of pointers really come into their own when we are dealing with arrays of strings. After all, an array of different strings is basically an array of arrays.

"MONDAY","TUESDAY","WEDNESDAY","THURSDAY","FRIDAY","SATURDAY","SUNDAY"

Here there are seven arrays of haracters of variable length strings. An effective alternative to using arrays of arrays (multi-dimensional arrays), as we have been looking at is to create an array of pointers to the strings. Now instead of changing, or swapping long strings around during sort routines, we can effectively, and simply, swap the pointers around in a pointer array.

The following program shows a pointer array being swapped around so that the string arrangement is different without actually moving the strings.

```
/* simple string sort using a pointer
array */

main()
|
static char val1[]="
ero";
static char val2[]="null";
static char val3[]="midd";
char *ptr[3]
int c,i
ptr[0]=val1;
ptr[1]=val2;
ptr[2]=val3,

  for (c=0, c<=2; c++)
     printf("%s n",prt[c])
     printf(" n");

  swap(&ptr[0],$ptr[1];&ptr[
     for (i=0 i<=2, i++)
        printf("%s_n",ptr[i]);

|
swap(x,y,
}
int *x,*y,*
;
{

int tmp;

  tmp = *x;
  *x = *y,
  *y = *
;
  *
 = tmp,

}
```

Some points to note in the forgoing program.

In order to pass the pointers to the pointer values correctly to the function SWAP we must use the address sign '&' in front of the pointer array values:

```
swap(&ptr[0], &ptr[1], &ptr[2]);
```

Also in the printf() I used the symbol %s which signifies a string (remember, %d = a denary (decimal) number, %c = a character).

Another aspect of using an array of pointers with string lists is how easily any string item can be located. This does create very efficient, fast, and short routines for string access. Here is a neat program to demonstrate this:

```
#include <stdio.h>
char *get_name(c)
int c;
{

static char *day[]={
                        "No      such
day","Monday"," uesday","Wednesday",

"Thursday","Friday","Saturday","Sunday"},

   return ((c<   c>7) ?day[0]:day[c]

   }

   main()
   {
   char c,

      printf("day number please <1 (-
7>\n\n");

       c=(getchar()-48);

     printf("%s\n",get-name(c));

   }
```

We start with the definition of a function of type char, which will return a pointer (one of the array pointers).

The pointer array is declared *day[], and since it is static, is assigned with the strings of the days of the week. After this, the last line of the function will return the selected string from the numeric input of the user. Let's look at this line in more detail.

```
return ((c<1 : c>7) ?day[0]:day[c])
```

If c is less than one c<1= or :: c is greater than seven c>7 then return day of

ero ?day 0] else return day of c :day[c].

Right, that is it for this month, lots to consider and practise. Next issue I shall be discussing, among other things, COMMAND LINE ARGUMENTS, or how to pass parameters to the program itself directly from the operating system by using two arguments, ARGC and ARGV. Plus our opening look at STRUCTURES.

Until then, bye...

TECHNO-INFO

More problems from our Agony Aunt, JASON FINCH, before he swans off on holiday to Germany (I must be paying him too much !!)

PRINTER CHIP

Dear CDU,
I have a couple of questions for you. Firstly, in the July issue you published an article on a Centronics Interface, but it refers to a Commodore BASIC listing (p30, lines 1 and 53). This does not seem to have been printed - or have I missed something? Second, in the August issue you published a letter about the MPS801 printer what is the modification ("special chip") giving true descenders and extra fonts? This would seem to remove one of this little machine's main shortcomings. If you can possibly help with these points I would be most grateful.
E.Coustick, Birmingham

DEAR MR COUSTICK
THE LISTING OF THE BASIC PROGRAM IS NOT PRINTED IN THE MAGAZINE - INSTEAD YOU WILL FIND THE PROGRAM FILED ON THE DISK GIVEN WITH THE JULY ISSUE. ON YOUR SECOND POINT, THE SPECIAL CHIP IS THE PRINTER IV CHIP THAT WAS ON THE MARKET A YEAR OR SO AGO. UNFORTUNATELY NO SUCH HARDWARE IS AVAILABLE TODAY, MOST LIKELY DUE TO THE FALL IN DEMAND FOR THE MPS801 PRINTER.

64 OR 128?

Dear CDU,
I understand that Commodore 64 programs on the disk can be run on the 128, but is the converse true? I think I have tried one or two labelled for the 128 on my 64 without success. Is there a way of doing it or is it impossible?
Alec Alkins, Devon.

DEAR ALEC,
THE PROGRAMS FOR THE COMMODORE 64 CAN ONLY BE RUN ON THE 128 PROPERLY WHEN THE LATTER MACHINE IS PUT INTO "64 MODE". THIS MEANS THAT THE 128 ACTUALLY EMULATES EXACTLY A 64 AND THAT IS WHY THE PROGRAMS ARE SAID TO BE ABLE TO RUN ON IT. THERE ARE ALSO A CERTAIN NUMBER OF PROGRAMS FOR THE 64 THAT ACTUALLY RUN ON THE 128 WHILST IT IS OPERATING AS A 128. THESE ARE MAINLY BASIC PROGRAMS THAT CONTAIN NO MACHINE CODE AND DON'T DO AN EXCESSIVE AMOUNT OF POKEING AROUND IN THE MEMORY. ANY OTHER PROGRAM WOULDN'T WORK BECAUSE OF THE DIFFERENT MEMORY MAPS FOR THE TWO MACHINES. THE MAJORITY OF PROGRAMS FOR THE 128 WILL NOT WORK ON THE 64 BECAUSE THE BASIC USED IN THE 128 HAS A NUMBER OF ADDED KEYWORDS THAT THE 64'S BASIC DOES NOT. MOST PROGRAMMERS UTILISE THESE EXTRA COMMANDS AND THE 64 SIMPLY DOESN'T UNDERSTAND THEM. IT IS BEST TO KEEP THE PROGRAMS WRITTEN FOR A 64 TO THE 64 AND PROGRAMS WRITTEN FOR A 128 TO THE 128.

WHY NO POUND SIGN?

Dear CDU
My set-up is a C64, two disk drives, 1901 monitor and a STAR LC10C printer, basically to suit the GEOS programs. I had spent hours trying to get the pound sign printed directly from the GEOS software without success (except to create a graphic character which meant transferring letters requiring this feature to GeoPaint with consequent editing problems). As you might expect, your reply to Matthew Langner in your September issue raised my hopes considerably so off I went full of hope. DIP switch number five on the LC10C gives English font, so it was switched off as required. The line you gave was entered and lo and behold, a pound sign! The only problem then being a reversal of upper and lower case fonts. Help! My printer produces NLQ font to a far better standard than the GEOS fonts which limits the effectiveness of the program. ESSL's suggestion was to buy their FontPack at twenty pounds which I thought a bit much. I hope that you can assist me in unravelling the confusion.
Glyn Hughes, Southampton.

DEAR GLYN,
YES YOU ARE RIGHT, THE FLICKING OF SWITCH FIVE DOES RESULT IN THE TWO CASES BEING SWAPPED BUT THIS IS JUST ONE OF THE SIDE EFFECTS THAT

YOU WILL HAVE TO LIVE WITH I'M AFRAID. I GRANT YOU IT'S A VERY BIG SIDE EFFECT BUT UNFORTUNATELY THERE IS NOT REALLY ANYTHING THAT CAN BE DONE ABOUT IT WITH GEOS. THAT IS THE ANNOYING THING ABOUT GEOS - IT'S AMERICAN AND THEREFORE DOES NOT INCORPORATE THE POUND SIGN. THE ONLY WAY FOR YOU TO GET EVERYTHING WORKING PERFECTLY WITH GEOS WOULD BE TO TYPE EVERYTHING THE WRONG WAY AROUND - CAPITALS WHERE YOU WANTED LOWER CASE AND VICE VERSA. I DON'T PERSONALLY THINK THAT BUYING THE FONTPACK WOULD RECTIFY ANYTHING IF YOU STILL WISH TO WORK IN NLQ. MY SUGGESTION TO YOU WOULD BE TO BUY A DIFFERENT WORD-PROCESSOR. THE WONDERFUL THING ABOUT GEOS IS THE VERSATILITY BETWEEN THE PACKAGES BUT IF YOU JUST WISH TO USE IT FOR A WORD-PROCESSOR IT IS, AS YOU SAY, NOT AS EFFECTIVE AS YOU WOULD WANT WHEN YOU USE NLQ. A NEW WORD-PROCESSOR WILL COST MORE THAN TWENTY POUNDS BUT WILL PROBABLY BE WORTH IT. I HOPE YOU ARE ABLE TO SORT SOMETHING OUT.

DISKS NEEDED!!

Dear CDU,
I am having some problems with formatting and copying with my 1571 disk drive and my C128 computer. I bought the C128 second hand and as a result I did not get the CP/M disks which I now understand are supposed to go with the machine. I was wondering if through your excellent column I could be put in touch with someone, or some firm, from whom I could purchase these disks.
W Nisbet, Plymouth

DEAR MR.NISBET,
I AM AFRAID THAT I'M NOT SURE WHETHER THE DISKS ARE STILL FOR SALE BUT I WOULD FIRST SUGGEST THAT YOU TRY A SHOP CLOSE TO YOU THAT SPECIALISES IN COMPUTER EQUIPMENT. FAILING THAT I CAN ONLY DO ONE OF MY FAMOUS PLEAS TO ASK IF THERE IS ANYBODY WHO KNOWS FOR DEFINITE FROM WHERE MR.NISBET COULD OBTAIN THE DISKS? SORRY I CAN'T BE OF ANY MORE HELP.

RASTERS AND SINE WAVES

Dear CDU,
Over the last few months I have become interested in the numerous effects that can be achieved by using the raster interupt facilities of the 64. I particularly like the effect where many different background colours are displayed at the same time, such as in Raster Editor in October's CDU. I also like the effect whereby different coloured bars move up and down the screen in such a way that they look as if they are moving in a circular fashion. I believe this is called "sinussing" but I don't understand how it is done. From the name it is obvious, to me anyway, that it has got something to do with sine waves but how does one make a bar move in this way? I know that some sort of table of values is set up but I haven't got a clue as to how to go about setting one up. Could you please shed a bit of light on this for me and possibly provide a sample program, as everyone else tells us the theory behind everything but nobody seems to want to reveal 100% of how to do anything. In case you need to know, I do understand assembly language and I am aware of the techniques involved to set up rasters and the awkward things like timings.
Edward Clarke, Bristol.

DEAR EDWARD,
THIS IS A BIT SPECIALISED SO IT'S A GOOD JOB THAT I KNOW A BIT ABOUT RASTERS. FIRST AND FOREMOST YOU ARE CORRECT IN THINKING THAT IT IS TO DO WITH SINE WAVES. IF YOU ARE MATHEMATICALLY MINDED, IMAGINE THE HORIZONTAL AXIS OF THE GRAPH OF THE WAVE AS BEING TIME RATHER THAN DEGREES AND THE VERTICAL AXIS BEING THE VERTICAL POSITION OF THE BAR RATHER THAN JUST A VALUE BETWEEN -1 AND +1. THEN YOU CAN SEE THE VERTICAL POSITION OF THE BAR AT ANY ONE INSTANT IN TIME. WHEN THE BAR IS NEAR THE TOP EACH INCREMENT IN VERTICAL POSITION HAS AN INCREASINGLY LONGER PERIOD OF TIME BETWEEN IT AND THE NEXT INCREMENT, UNTIL WE START GOING BACK DOWN AGAIN. THEREFORE THE BAR APPEARS TO SLOW DOWN AND THEN GRADUALLY SPEED UP ON THE WAY DOWN, BEFORE AGAIN SLOWING WHEN IT REACHES THE "TROUGH" OF THE SINE WAVE. THERE IS A SIMPLE ALGORITHM FOR RELATING ALL THIS TO COMPUTERS AND RASTER BARS. A TABLE OF FIGURES DOES INDEED NEED TO BE SET UP AND THESE REPRESENT THE VERTICAL POSITION OF THE BAR AT ANY ONE INSTANT - THERE ARE NO VARIABLE TIME DELAYS. IT'S JUST THAT THE BAR IS DISPLAYED AT THE SAME POSITION FOR MANY CONSECUTIVE CALLS OF THE INTERRUPT ROUTINE. IT IS VERY DIFFICULT FOR ME TO SUMMARISE EVERYTHING ABOUT SINUSSING IN THIS SHORT SPACE BUT ANYWAY, HERE IS THE ALGORITHM. Z= INT(0.5*PR* SIN(6.283185*NM/VL) +PR/2) WHERE PR IS THE VERTICAL RANGE OF THE MOVEMENT IN PIXELS. VL IS THE TOTAL NUMBER OF VALUES IN THE TABLE, AND NM IS THE NUMBER OF THE VALUE IN THE TABLE. THE CONSTANT 6.283185 SIMPLY INVOLVES THE COMBINATION OF THE CONSTANT PI/180 TO CONVERT TO DEGREES AND 360/VL WHICH CALCULATES THE NUMBER OF STEPS IN DEGREES ALONG THE WOULD-BE HORIZONTAL

AXIS OF THE WAVE. THE VALUES OF PR/2 ARE REQUIRED BECAUSE YOU WANT TO START IN THE "MIDDLE" OF THE VERTICAL AXIS BUT YOU WANT THE LOWEST VALUE TO BE ZERO AND NOT A NEGATIVE NUMBER, AND THAT IS WHY THE PR/2 IS ADDED AT THE END. ALL THAT IS A BIT COMPLICATED BUT THERE IS A PRACTICAL USE OF IT ON THIS ISSUE'S DISK FILED AS "PROB1A". THE PROGRAM ASKS YOU FOR THE NUMBER OF VALUES TO BE CREATED AND THE RANGE OVER WHICH THE BAR WILL TRAVEL. IT THEN DISPLAYS THE NECESSARY VALUES AND MOVES A SPRITE ON THE RIGHT OF THE SCREEN TO SHOW YOU WHAT IS HAPPENING. ALSO ON THE DISK IS "PROB1B" WHICH IS AN ASSEMBLY LANGUAGE PROGRAM FORMING A PRACTICAL EXAMPLE OF THE INTERRUPTS IN OPERATION SHOWING EXACTLY HOW YOU HAVE TO ACCESS THE SINUS TABLE AND IMPLEMENT THE INFORMATION WITHIN IT. NOTE HOW, IN PARTICULAR, THE POSITION OF THE RASTER LATCH IS KEPT CONSTANT. YOU MAY CHANGE THE VALUES IN THE COLOUR TABLE IF YOU WISH - FOR A TOATALLY BLACK BACKGROUND MAKE THEM ALL $00 BYTES. I APPRECIATE THAT ALL THIS IS A BIT COMPLICATED ON FIRST READING, BUT EVERYTHING SHOULD SORT ITSELF OUT ONCE YOU HAVE FAMILIARISED YOURSELF WITH THE PROGRAMS.

PROBLEMS IN S.A.

Dear CDU,
The problem starts with the fact that I live in South Africa and that 'new' software is not available. Therefore I have only programs dated to 1987. I bought a tape version of "Game Over" and I have difficulties in getting to the second world. Seeing as how it's my best game, I would like to know what the password is to the second stage of the game. I have also connected a reset button to the user port. Are there any POKEs and SYSs I can use to gain infinite lives and so forth? I would also appreciate it if you could tell me what games in CDU in the past have used the SEUCK. The only one I can think of is Atlantis.
Colin Sanderson, South Africa.

DEAR COLIN,
I DO NOT OWN A COPY OF THE GAME THAT YOU MENTION AND THEREFORE CANNOT OFFER ANY HELP ON HOW TO GET TO THE SECOND STAGE OF IT. I WOULD SUGGEST THAT YOU WRITE TO ONE OF THE OTHER COMMODORE MAGAZINES, BUT ONE THAT SPECIALISES IN PERHAPS GAME REVIEWS. SOMEONE THERE IS BOUND TO BE ABLE TO FIND OUT THE INFORMATION THAT YOU WANT THERE. AS FOR GAMES THAT HAVE BEEN PROGRAMMED USING THE SHOOT-'EM-UP CONSTRUCTION KIT, I CAN THINK OF ONLY TWO OTHERS. THOSE BEING COBALL IN VOLUME 2 ISSUE 6 (SEPTEMBER/OCTOBER 1989) AND B-RAID IN VOLUME 3 ISSUE 1 (NOVEMBER 1989). IF THERE WAS ANOTHER ONE OR POSSIBLY TWO THEN MY APOLOGIES GO TO THE AUTHOR FOR OVERLOOKING IT!

GIVE US A MAG JOHN

Dear CDU,
Just one problem. I recently acquired the four complete volumes of the serialised INPUT magazine, or so I thought I had. They were published by Marshall Cavendish Ltd. Unfortunately they clear the shelves of all back issues six months after publication. Volume 4, number 40 is the one I am seeking. Has anybody out there got a copy of it? I am willing to pay for it, and would be eternally grateful for any help on this matter.
L.C Sparrowhawk, Crawley.

DEAR MR.SPARROWHAWK,
THANKS FOR YOUR LETTER. ALL I CAN DO IS TO SAY THAT IF ANYONE CAN HELP OUT THEN PLEASE WRITE TO TECHNO INFO WITH ANY FURTHER DETAILS THAT YOU MAY CONSIDER NECESSARY. I'LL KEEP YOU POSTED OF ANY DEVELOPMENTS.

BRIDGE PROGS GALORE

Dear CDU,
I have been a reader of CDU from the very first issue and think it is a very good magazine. Regarding the query from Mr.Wright of Shropshire in the October issue, I have a game of Bridge on a game package called "The Complete Home Entertainment Centre". As well as Bridge it has Mah Jong, Dominoes, Wordsearch, Pinball and Video Card Arcade. This complete package was produced by CDS Software Ltd, CDS House, Beckett Road, Doncaster, South Yorks, DN2 4AD. Their telephone number is given as 0302-21134. I do not know if this is the present address but hopefully Mr.Wright will be able to get a copy of the program.
W.M Pusey, The Isle of Wight.

DEAR MRS PUSEY,
THANKYOU VERY MUCH FOR TAKING THE TIME TO WRITE THE LETTER INFORMING ME OF THE PACKAGE. HOPEFULLY NOW MR WRIGHT WILL BE ABLE TO FOLLOW THIS ONE UP. I MUST ALSO SINCERELY THANK THE MANY OTHER READERS WHO SENT IN SIMILAR INFORMATION SO PROMPTLY. I HAVEN'T GOT ROOM TO MENTION ALL YOUR NAMES, BUT YOU KNOW WHO YOU ARE.
EDITORS NOTE - IN FACT WE HAD NO LESS THAN 17 PEOPLE OFFERING THE SAME HELP. THANK YOU ALL!!

LACK OF UNDERSTANDING

Dear CDU,
Being very new to computers I understand very little of the jargon used and find the instructions used in computer magazines somewhat confusing (may be I'm thick or just too old!). I have just bought my first copy of CDU (Volume 3 Number 12) and have had trouble with the programs. Firstly, saving Spread-Ed files and using them on Graph-Ed. Mention is made of .SHT, .GRA and .STAT extensions. What are they and how do I use them. My next problem. I am using a C64, a 1541II drive and an MPS 1550C printer, but it will not work properly. Sometimes disks will not load and I have programs with PRINT commands and when used the screen shows it is printing but the printer is doing nothing. Even the test disk that comes with the 1541II has a printer test program, but when loaded it just states "Device not present". Am I doing something wrong or is the printer incompatible?
W Hunter, Newcastle

DEAR MR.HUNTER
FIRSTLY WITH RELATION TO THE SUFFIXES THAT YOU MENTIONED, IN THE INSTRUCTIONS TO THE PROGRAMS THERE ARE DETAILS OF HOW TO LOAD AND SAVE INFORMATION FROM THE SPREADSHEET AND THE GRAPH PLOTTING PROGRAM. WHEN IT SAYS THAT YOU MUST USE A SUFFIX THEN SIMPLY ADD IT TO THE NAME THAT YOU ENTER FOR THE FILE WHEN IT IS SAVED OR LOADED. WHEN IT TELLS YOU TO OMIT THE SUFFIX THEN JUST IGNORE IT. ALL THE SUFFIXES DO IS TELL YOU THAT A PARTICULAR FILE ON A DISK IS TO BE USED WITH EITHER THE SPREADSHEET OR THE GRAPH PLOTTER. THEY ALSO HELP TO ENSURE THAT THE FILE THAT YOU WANT TO LOAD, IS CAPABLE OF BEING LOADED INTO EITHER PROGRAM. THAT IS, THAT A FILE DESIGNED FOR THE GRAPH PROGRAM WILL STORE DATA DIFFERENTLY TO ONE THAT IS DESIGNED ONLY FOR USE WITH THE SPREADSHEET PROGRAM AND THEY THEREFORE HAVE DIFFERENT SUFFIXES TO PREVENT THE WRONG DATA BEING LOADED. WITH REFERENCE TO THE PRINTER, I THINK THAT IT IS COMPATIBLE WITH THE 64 AND THEREFORE CAN ONLY THINK THAT YOU MUST HAVE OVERLOOKED SOMETHING THAT YOU THOUGHT WAS TOO OBVIOUS. THE DEVICE NOT PRESENT ERROR INDICATES TO ME THAT THE PRINTER IS POSSIBLY SET UP AS DEVICE NUMBER FIVE - IT SHOULD BE NUMBER FOUR. HAVE A LOOK IN THE MANUAL TO SEE HOW TO CHANGE THE DEVICE NUMBER AND IF IT IS SET TO FIVE THEN CHANGE IT. YOU CAN PERFORM A SIMPLE CHECK YOURSELF FIRST FROM POWER-UP OF THE COMPUTER. TYPE: OPEN 1,4: PRINT#1,"HELLO": CLOSE 1. IF THAT PRODUCES AN ERROR OR DOES NOTHING, TRY: OPEN 1,5: PRINT#1,"HELLO": CLOSE 1. IF THAT STILL PRODUCES AN ERROR THEN THERE MAY BE A PROBLEM WITH THE 'DAISYCHAINING'. CHECK THAT ALL THE LEADS GO TO WHERE THEY SHOULD. THE PROBLEM WITH THE DRIVE SOUNDS RATHER ODD - HAS THE PRINTER GOT AN ON-LINE BUTTON? IF SO, MAKE SURE THAT THE PRINTER IS ON LINE (OR SIMPLY READY TO PRINT) BEFORE STARTING THE LOADING. I HOPE THAT I HAVE BEEN OF SOME HELP TO YOU.

QUOTES ON STAR LC10C

Dear CDU,
In the July 1990 issue there was a query on using quotes with a word processor. If the man's STAR LC10C is like the mono version it is possible. The printer can download the character ROM into RAM and then you can change a character not needed to quotes. The first program that I have enclosed is for draft mode and the second is for NLQ mode. Lines 110 to 140 set up the changing of ROM to RAM and line 150 tells the printer which character to change (character 33 is an exclamation mark). The main part of the programs were taken from the printer manual. I hope he finds them helpful.
L.J.Todd, Bedfordshire

DEAR MR.TODD,
THANKS VERY MUCH FOR SENDING THE PROGRAMS. I HAVE INCLUDED THEM ON THIS ISSUE'S DISK - THEY ARE FILED AS 'TECHNO1' AND 'TECHNO2' FOR DRAFT AND NLQ RESPECTIVELY. MAY I JUST ADD THAT IF YOU SHOULD LOAD AND RUN THE PROGRAMS IMMEDIATELY BEFORE LOADING THE WORD-PROCESSOR AND THE PRINTER SHOULD NOT BE RE-INITIALISED IN BETWEEN DOWNLOADING THE CHARACTER AND PRINTING IT. ALSO IF YOU ARE USING NLQ, MAKE SURE THAT YOU HAVE SELECTED YOUR DESIRED NLQ FONT BEFORE RUNNING THE PROGRAM.

MORE LACK OF KNOWLEDGE

Dear CDU,
I am not that much of a super computer whiz, but I am not entirely stupid either so could you please give more in depth details of those more complicated utilities and programs. I have had my computer for under two years and I am not quite familiar with all the computer terms. For example the Base-Ed instructions were only complicated because I am not sure what it means by "fields" and "strings", and suggest that you explain a little more. I look forward to your reply and not by post, but in next month's issue.
Paul Donnelly, Huyton.

DEAR PAUL,
FIRSTLY, AS YOU REQUESTED, I HAVE PUBLISHED YOUR LETTER AND MY REPLY FOR ONE AND ALL TO SEE. SECONDLY I MUST STRESS THAT IT IS ENTIRELY UP

TO THE AUTHOR OF THE PROGRAMS TO SUPPLY SUITABLE INSTRUCTIONS BECAUSE WE SIMPLY HAVEN'T GOT THE TIME HERE TO BE ADDING THINGS IF WE FEEL IT IS NEEDED. I HAVE SENT OUT PLEAS IN THE PAST FOR PROGRAMMERS TO REVEAL EVERY DETAIL NECESSARY IN THEIR INSTRUCTIONS BUT JUST FOR YOU I SHALL DO IT AGAIN ALL PROGRAMMERS PLEASE REVEAL EVERYTHING THAT YOU CAN ABOUT THE USE OF THE PROGRAM. NOW TO ANSWER YOUR QUERY AS TO WHAT "FIELDS" AND "STRINGS" MEAN WHEN YOU USE A DATABASE YOU ARE LIKELY TO GROUP THINGS UNDER HEADINGS THE SIMPLEST ANALOGY BEING THE ADDRESS BOOK STYLE WHERE YOUR HEADINGS WOULD BE "NAME" "ADDRESS" AND "PHONE NUMBER" AMONGST OTHERS EACH OF THESE IS CALLED A FIELD THEREFORE YOUR DATABASE CONTAINS THREE FIELDS OTHER EXAMPLES OF FIELDS COULD BE "VIDEO NAME" "COMPANY", "MODEL" AND SO FORTH ANY HEADING SUCH AS THAT IS CALLED A FIELD ONE PIECE OF INFORMATION IS TAKEN FOR EACH FIELD AND THAT MAKES A RECORD THEREFORE ONE ENTRY (RECORD) IN A DATABASE COULD HAVE "JASON FINCH" STORED IN THE FIELD WITH TITLE "NAME" "COOK CLOSE, RUGBY" STORED IN THE FIELD WITH TITLE "ADDRESS" AND "573080" STORED IN THE FIELD WITH NAME "PHONE NUMBER" THAT INFORMATION MAKES UP ONE RECORD AND YOU CAN HAVE AS MANY AS THOSE AS THE DATABASE WILL PERMIT EACH OF THOSE THINGS THAT IS STORED IN THE MEMORY THAT I HAVE SHOWN BY ENCLOSING THEM IN QUOTE MARKS IS CALLED A STRING ANYTHING THAT YOU ENTER THAT CONSISTS OF LETTERS AND NUMBERS IS CALLED A STRING THEREFORE EACH RECORD SHOULD HAVE A STRING STORED UNDER EACH FIELD HEADING I HOPE THAT I HAVE NOT SIMPLY SERVED TO ADD TO THE CONFUSION!

TROUBLE WITH B.O.S.S.

Dear CDU
I am having some difficulty with loading B.O.S.S. on the September disk. Upon loading the program from the menu you are presented with a facility for moving a sprite around or looking at a map of an island. No commands anywhere. Listing the program offers about twenty lines of BASIC confirming the sprite and island bit, but nothing else. What can I do? I have tried loading it and typing one or two of the commands listed with no result. Please try to shed some light on this problem.
M McGrail, Manchester.

DEAR MR MCGRAIL,
THE MAIN PROGRAM IS ACTUALLY JUST A MACHINE CODE FILE, THE PROGRAM THAT LOADS FROM THE MENU BEING A DEMONSTRATION OF THE CODE IN ACTION. YOU MUST LOAD THE PROGRAM BY TYPING LOAD"THE BOSS",8,1 FOLLOWED BY NEW AND THEN SYS49152 OR ALTERNATIVELY TO LOAD THE CODE FROM WITHIN THE PROGRAM YOU SHOULD CONSULT THE DEMO THAT I HAVE PREVIOUSLY MENTIONED. THE SYS49152 COMMAND SWITCHES ON THE SOFTWARE ALLOWING YOU TO ENTER THE COMMANDS YOU SHOULD NOW BE ABLE TO DO WHATEVER YOU WANT REMEMBERING TO PREFIX EACH WITH AN APOSTROPHE MARK. FOR EXAMPLE, AFTER SWITCHING YOUR COMPUTER ON, TYPE THE FOLLOWING LOAD"THE BOSS",8,1 FOLLOWED BY NEW WHEN IT HAS LOADED THEN TYPE SYS49152 FOLLOWED BY 'CLS THAT SHOULD CLEAR THE SCREEN I HOPE THAT YOU WILL NOW BE ABLE TO GET EVERYTHING TO WORK

TIP OF THE MONTH

Well, letters out of the way and now onto this month's handy tip! I have received quite a few which pleases me no end but the one that I promised you last month is here this month - it comes to you courtesy of Mr R.Smedley from Sutton-in-Ashfield, Notts. and takes the form of a machine code routine which you will find on this issue's disk. To load it simply enter LOAD"TECHNO TIP",8,1 Take it away ..

THE PROGRAM WILL ALLOW THE INDIVIDUAL TO CONTROL THE SPEED AT WHICH C64 BASIC OPERATES THE PROGRAM IS ACTIVATED BY SYS49152 TO CHANGE THE SPEED AT WHICH BASIC OPERATES YOU THEN PRESS THE COMMODORE KEY AND THE CONTROL KEY SIMULTANEOUSLY. THIS MUST BE DONE WHILST THE COMPUTER IS EXECUTING A BASIC PROGRAM) AT WHICH POINT THE COMPUTER WILL STOP WHAT IT IS DOING AND PROMPT YOU TO ENTER A NEW SPEED BY PRESSING ONE OF THE NUMBER KEYS (ZERO IS NORMAL SPEED AND NINE IS COMPLETELY PAUSED) THE COMPUTER WILL THEN CONTINUE FROM WHERE IT LEFT OFF. THE PROGRAM OCCUPIES LOCATIONS 49152 TO 49824 (IT ADDITIONALLY USES MEMORY UP TO LOCATION 50431) AND DOESN'T USE THE INTERRUPTS, SO IT SHOULD BE COMPATIBLE WITH MOST OTHER PROGRAMS. THE PROGRAM WAS DESIGNED TO AID THE DEBUGGING OF BASIC PROGRAMS BY PROVIDING THE FACILITY TO SLOW THEM DOWN TO SUCH A POINT THAT YOU COULD SEE EXACTLY WHERE THE BUGGED PROGRAM FAILS

Thanks very much Mr.Smedley - I am sure that plenty of people will find it very useful. Anyway, that concludes this month's utterings from me and so it is just left to say that if you have a problem, and nobody else can help, and you can find them (and you've got enough money), then why not call the Techno Info team! He can be reached at the following address: CDU Techno Info, 11 Cook Close, Brownsover, Rugby, Warwickshire CV21 1NG.

SCREEN DESIGNER (UPDATE)

Those veritable gremlins have been at it again.

WILLIAM CHRISTIE

In the MAY issue of CDU we published an excellent utility for designing your own screens and platforms for games. Unortunately, it has come to my attention that there is a couple of minor bugs in the program SCREEN DESIGNER, we present the updated version for your pleasure on this months disk.....PAUL EVES (Editor)

Whilst using the SCREEN DESIGNER module from the above mentioned utility there are two problems you may come against. The first concerns the use of the "XY" function, from the F7 menu. This function was the last one that I developed and use was made of the ACTION REPLAY MK V cartridge. As this cartridge allows HEXADECIMAL to be used within Basic I made use of this number base to enable me to keep track of vital memory locations whilst in the Monitor mode. I completely forgot to convert the two numbers in line 2400 of the "XY" routine.

The second problem was seen with the use of the "LOAD" function, also from the F7 menu. Here, use was made of location #1022 to determine whether screen data is currently present in memory or not. The problem being that when use is made of this function to load in screen, character data or the directory, this location is set to 1. (This tells the computer that screen data is present). However, I didn't notice that this location is also set if character data or the the directory is loaded - and this is where the problem lies since new screens are now stored in the location pointed to via the vector at $4000/1.

Both these problems have now been corrected, and on the disk you will find the new version. I apologise to you all for any inconvenience this may have caused you

PLEASE NOTE: When running/testing a game you should reset Basic back to $0801 AFTER running SPDRIVER BAS (entry program) and BEFORE loading in PLAT.BAS by typing SYS58260, otherwise PLAT.MC will overwright PLAT.BAS

C64 AMIGA C128

ONLY £149.00 P&P £1.95

C64 DISK DRIVE

COMMODORE 1541C II DISK DRIVE, SLIMLINE CASE AND EXTERNAL POWER SUPPLY FAST ACCESS OF SOFTWARE WITHOUT PROBLEMS

PLUS FIVE FREE DISKS

THE ONLY DRIVE 100% COMPATIBLE WITH THE 64 128

FREE GEOS

INCLUDING GEOS WORD PROCESSOR PAINT PACKAGE [illegible]

SPECIAL XMAS OFFER!

FREE 8 GAMES

SECRET AGENT
NIGHTBREED
[illegible]
RESISTANCE
TRIVIAL PURSUIT
LONE ZION
SPLIT PERSONALITIES
SHADOW WARRIORS
[illegible]

NEW AMIGA A500 SCREEN GEMS £379.00 £5.00 P&P.

THE TOP SELLING COMPUTER WITH STEREO SOUND, ARCADE GRAPHICS AND STUNNING COLOUR. THE SCREEN GEMS PACK INCLUDES 4 GREAT GAMES: DAYS OF THUNDER ● NIGHTBREED ● BACK TO THE FUTURE 2 ● SHADOW BEAST ● DELUXE PAINT 2

3.5" DISK DRIVE, MOUSE AND TV MODULATOR ARE ALL INCLUSIVE

C64 PRINTERS

COMMODORE MPS1230	£149.00
SEIKOSHA SP 180VC	£159.00
CITIZEN 120D+	£149.00
CARRIAGE	£3.50
C64 MOUSE	£26.50
C64 POWER SUPPLY	[illegible]
C64 DATA RECORDER	£24.50
P&P	£1.95

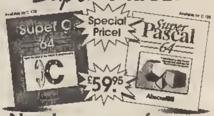
C64 COMPUTER NIGHT MOVIES

INCLUDES 64C COMPUTER DATA CASSETTE, 2 JOYSTICKS, 8 GREAT GAMES (USE DISK DRIVE) PLUS INTRO AUDIO TAPE

£140.00 + £5.00 P&P

C.M.S. [illegible] 45 WHITBREAD ROAD BROCKLEY LONDON SE4 2BD 081 469 2346

Super COMBO Super PRICE!



New Languages for your C-64 and C-128

Super Pascal - 64	£39.95	Super Pascal - 128	£39.95
Super C - 64	£39.95	Super C - 128	£39.95
Cobol 64/128	£29.95	Fortran - 64	£29.95
ADA-64	£19.95	Forth - 64	£12.95
LOGO - 64	£14.95	Pilot-64	£14.95
Video Basic - 64	£19.95	Pascal - 64	£14.95
Zoom Pascal	£9.95	Assembler/Monitor - 64	£9.95

SPECIAL OFFER Super C + SuperPascal (C64 or C128) £59.95

Send S.A.E. for full details

ADAMSOFT, 18 NORWICH AVENUE, ROCHDALE, LANCS OL11 5JZ
Access/Visa orders Tel: 0706 524304

KANGAROO KORNER

CDU presents a first for the magazine. We have a FEMALE reader, who just happens to be Austrailian, that has sent us a suite of useful routines for your enjoyment

Over the next few issues, we will be presenting all six of these routines. They are BETTER BACKUPS, CUSTOM BASIC, SPACE INSERTER, UNIVERSAL VERIFIER, LOAD ADDER and PERMANENT HEADER IRQ. Roughly they do the following:

BETTER BACKUPS - Solves some of the problems encountered when Freezing large games.
CUSTOM BASIC - A bit of a fun program that allows you to play around with the Basic keywords.
SPACE INSERTER - A nice way of saving space in your code when you have a lot of printing to do.
UNIVERSAL VERIFIER - A simple method of getting around the restricted use of VERIFY
LOAD ADDER - Allows a normal load to be added to an ACTION REPLAY CARTRIDGE preloader.
PERMANENT HEADER - An easy way of constantly displaying a message on the top of the screen.

I hope you find these little routines as useful as I have. In this issue I bring you the first of these, also the longest, namely BETTER BACKUPS. On the disk you will see the source code for this utility. Play around with it to your hearts content. Try a few experiments, but above all, enjoy yourself.

ELAINE FOSTER

BETTER BACKUPS

This article presents a method for ensuring that freezer-copied files actually work as desired, and incidentally presents some nice autoloading or autobooting Machine Code routines which you can use.

You may backup an autorunning game from tape to disk, or disk to disk, using a freezer device, but when it is very long the copy may not load properly. Here are some solutions to that problem. Becaues most games use Machine Code, these ideas use Machine Code, but anyone with an ordinary Monitor program, or Cartridge monitor, may apply them, and will also find some powerful tools for using the Kernal ROM.

The problem may occur when a Freezer device breaks a program, usually a game of over 200 blocks in length, into two or more files on the disk: a loader (which may put sprite data into CIA locations), and one or more clones. The main trouble is that this backup which was saved by the Freezer may not load when using fastloading hardware or software. And who, these days, would use a 1541 drive at its "normal" speed? There are also other problems with such backups, and these will be discussed here. Freezer cartridges have their limits

The first sign of the problem is when you try to load the copied files, and you see a "FILE NOT FOUND" error. If that happens, look up this article and try these ideas.

FINDING BA AND EA

First load the shorter of the copied files, but do NOT run it. These ideas will only apply to such programs; self-running ones require more complicated measures. Then find its Beginning Address (BA) and Ending Address (EA). Your Freezer cartridge, and some Monitors, will usually tell you that during the load If not, look at locations 43 to 46.

```
  PRINTPEEK(43)+256*PEEK(44)
```
gives you the Beginning Address (BA)
```
  PRINTPEEK(45)+256*PEEK(46)
```
gives you the Ending Address (EA).

Alternatively you can use your Monitor.
```
.M 002B
```
which will give a string of 8 numbers in Hex. In general hexadecimal numbers are preceeded by '$', but in the Monitor that only happens in Disassembly " M" listings only show the number, and you have to realise that they are in hex. $2B and $2C are low byte (LB) and high byte (HB) for the BA, and $2C and $2D are LB and HB for the EA. In general, for decimals,
```
  ADDRESS = LB+256*HB
```

THE KERNAL LOAD ROUTINE

To find whether the loader file is doing its job properly you will need to understand how Machine Code (MC) loads a program. It is wordy, but not difficult. In Basic you merely type;

```
LOAD"progname",8,1
```

the Basic Interpreter does the rest. In MC this requires three subroutines which are found in the KERNAL which is located from $F000 (57344) onwards. It is the 'BRAIN' which directs all INPUT/OUTPUT operations.

1. SETLF: 'Set logical file number, device number and secondary address'. Here is an example. Using your Monitor enter the following instructions, beginning at hexadecimal $2735 (10037 decimal). By the time you finish this article you will have entered a working program which can load others by Machine Code (MC). The actual addresses are not shown in the following lists, only the mnemonic instructions. In practice you would enter;

```
.A 2735 LDA #$08
```

for the first line, press RETURN, then merely enter each of the other lines in turn, as shown. The Monitor takes care of the assembly and keeps track of the addresses. Do NOT enter the semicolons or comments; they are for your information only, and would confuse the Monitor.

```
LDA #$08  ;file number 8
LDX #$08  ;device number 8
LDY #$01  ;secondary address 1
JSR $FFBA ;SETLFS subroutine
```

LDA #$08 means 'LOAD 8 Into the Accumulator", and similarly for the X and Y registers. Therefore 8 is put into the Accumulator, 8 into the X register and 1 into the Y register. JSR is similar to GOSUB, and calls the SETLFS kernal subroutine which uses those values of A,X and Y to record the file parameters.

2. SETNAM; "Set Filename Parameters". For example, say that the name of the file to be loaded is "GAME", and that the ASCII numbers for that name are at location $274C. Reasonably, the high byte of that is $27 and the low byte is $4C. So, now continue the assembly on your Monitor (again without semicolon or comments!)

```
LDA #$04  ;four characters in file name
LDX #$4C  ;low byte of filename
LDY #$27  ;high byte of filename
JSR $FFBD ;call the SETNAM routine
```

Now put the ASCII number for "GAME" at $274C; Enter .M 274C, and the result will be a string of 8 numbers, and at the right side of the screen, their character equivalents, if any. Put the cursor on the first of the character places at the right, and type GAME and press return. You will then see the first four numbers in the line as 47 41 4D 45, the hexadecimal ASCII for GAME. (In some Monitors you will need to enter the numbers only: look them up in any ASCII table).

The X and Y registers are 'pointing' to this location. They tell SETNAM at $FFBD that the name is there, and the A register tells it how many characters there are in it.

3. LOAD; "Load RAM from a device". For example, add this to the assembly:

```
LDA #$00  ;0=LOAD, 1=VERIFY
JSR $FFD5 ;Call the load routine
```

The secondary address (SA) of SETLFS (Item 1, above) is very important for LOAD. If it is 0 the new porgram is loaded into RAM beginning at a location defined by X (LB) and Y (HB). If SA=1 the load is determined by the header information at the beginning of the first data block on the disk. It is not 'relocated'. If in doubt, try SA=1. All of this, so far, has been equivalent to the Basic statement; LOAD"GAME",8,1

We shall return to this routine again, so for now, save it to a formatted disk (NOT the CDU one). On your Monitor enter,

```
S"LOADER/1",08,2735,2750
```

(which assumees the saving format: DN,BA,EA. If your Monitor uses another, of course use that instead).

IS THE WRONG FILE BEING LOADED

Now, let us look at some of the common sources of the non-working first file of the game. Load that first program into memory, and find BA and EA as described above. Then use the Monitor to find the LOAD subroutine. Say that BA=$0801 and EA=$2734 (these were the actual ones in my problem). Then, realising that LOAD is located at $FFD5, and that the microprocessor always treats this in the order LB, HB, we can hunt for JSR $FFD5;

```
.H 0801 2734 20 D5 FF
```

This will give you the addresses where that combination 20 D5 FF occurs in the range 0801 to 2734. The $20 stands for "JSR". If you do not find anything, try;

```
.H 0801 2734 4C D5 FF
```

where the $4C stands for "JMP" (GOTO in Basic). Or the $20 or $4C could be left out and you could merely hunt for D5 FF, as long as it did not give too many spurious results.

Now look at each of the locations found, and scroll upwards with your cursor to see what preceeds JSR $FFD5 (or JMP $FFD5). You ought to see LDA #$00 just above it, and before that the JSR $FFBD and JSR $FFBA groupings of SETNAM and SETLFS. (If you don't see either, skip to "IS NO FILE BEING LOADED" part of this article). Locate JSR $FFBD and let the X and Y registers tell you where the file name is located ($274C in the above example). Use the "M" command of the Monitor to tell you what those letters are. If they are not the same as the name of the next file this loader is trying to load, change them! If the new name is of a different length, adjust the LDA of the SETNAM routine as necessary. Then save the changed program,

```
.S"new filename",08,0801,2734
```

Now run the modified loader program and it should work well. In the example shown you would do this by leaving the Monitor and entering the command SYS10037 from the keyboard. (10037 is the decimal of $2735: Your Monitor can give you that translation).
Only one problem may arise. If the new name is longer than the old, it may not fit in the space available. In that case simply place it as the end of the program and change the SETNAM X and Y pointers accordingly, and the EA for saving.

IS THE WRONG DEVICE NUMBER USED

In this case, all is well, but the device number of SETLFS is 1 (for tape) instead of 8 (for disk), which is not surprising if the copy came from a tape. In that event, locatae the SETLFS area, and change LDX #$01 to LDX #$08, and save as before. This will usually not occur on the better Freezers.
A complication is possible though. TAX means "transfer contents of A to X". You might find;

```
 LDA #$01  ;file 1
 TAX       ;device 1
 LDY #$01  ;SA 1
 JSR $FFBA ;SETLFS
```

The solution is simple. Change the first line to LDA #$08, because the file number is not important. If the third line is TAY (which use only one byte of code) however, you are in trouble, because there is no room to insert an LDY (which uses 2 bytes). You cannot insert "lines" as in Basic; everything is crowded together. In that event you might jump to another location, but you would have to know exactly what you were doing. Machine Code can be very powerful, but it can also create some powerful catastrophes, and is much harder to debug than Basic.

IS NO FILE BEING LOADED

There is one other possibility, and it is most annoying. You freeze your game, but it will only load if you do not use a fast-loader. You can lie down and have a nice nap before 230 blocks have loaded without one! This happens because the copy only uses LOAD, and neither SETLFS nor SETNAM is included at all. The reason why this works is rather complicated and will not be discussed here. But the cure is that you have to add the preliminary routines. That is what happended tor the game which started this whole enquiry. But MC is, as mentioned, not like Basic where you can add lines between. So, you must add them at the end, and use the equivalent of a GOTO to access them. In this case, BA was $0801 and EA was $2734. So, the extra routine was added at $2735. A disassembly of the relevant part of the original routine is shown here, and you can see that SETLFS and SETNAM are absent from all lines preceeding $0835 where JMP $FFD5 occurs. For the advanced reader it also shows why LOAD alone actually works without the others (X=0 and Y=0 when entering the subroutine at $0838, and the loaded name is the same as the previous one, proceeded by "1"). It is a clever idea, but useless, because fastload corrupts $BB

```
.> 080D 78            SEI
.> 080E E6 01         INC $01
.> 0810 A2 1C         LDX #$1C
.> 0812 A0 00         LDY #$00
.> 0814 B9 47 08      LDA $0847,Y
.> 0817 99 00 D0      STA $D000,Y
.> 081A C8            INY
.> 081B D0 F7         BNE $0814
.> 081D EE 16 08      INC $0816
.> 0820 EE 19 08      INC $0819
.> 0823 CA            DEX
.> 0824 D0 EE         BNE $0814
.> 0826 C6 01         DEC $01
 > 0828 20 38 08      JSR $0838
.> 082B C8            INY
.> 082C 84 B9         STY $B9
.> 082E A9 08         LDA #$08
.> 0830 48            PHA
.> 0831 A9 0C         LDA #$0C
.> 0833 48            PHA
.> 0835 4C D5 FE      JMP $FFD5
.> 0838 A5 BB         LDA $BB
.> 083A D0 02         BNE $083E
.> 083C C6 BC         DEC $BC
.> 083F C6 BB         DEC $BB
.> 0840 E6 B7         INC $B7
.> 0842 A9 31         LDA #$31
.> 0844 91 BB         STA ($BB),Y
.> 0846 60            RTS
```

The absence of $FFBA or $FFBD, was discovered in the first instance by hunting for them using the monitor,

```
.H 0801 2734 BA FF and
.H 0801 2734 BD FF and finding nothing.
```

To cure this, the entry at $0835 was changed to

```
.> 0835 4C 35 27 JMP $2735
```

which simply told the computer to jump to the new LOAD routine instead of the to LOAD. The new routine was then located at $2735, just above the old EA, and which looked like this (the content of "LOADER/1" as above).

THE LOAD ADDITION

```
LDA #$08 ;file number
LDX #$08 ;device number
LDY #$01 ;SA for nonrelocating load
JSR $FFBA ;SETLFS
LDA #$04 ;number of characters
LDX #$4C ,LB of $274C, location of filename
LDY #$27 ,HB of $274c
JSR $FFBD ;SETNAM
LDA #$00 ;Load, not verify
JMP $FFD5 ;Load
```

and "GAME" is entered at $274C as described above, using the "M" command of the Monitor. And that is all. The old BA was $0801 and the new EA was $2750 (one more than the last address used by this added routine), so the whole game loader was saved as,

```
.S"new filename",08,0801,2750
```

Discerning readers can see that it would be possible to substitute TAX for the LDX #$08 in the second line above, but that will reduce the length by one byte, requiring the name location to be changed, etc.

TRY IT YOURSELF

All of this takes many more words to explain than to do. Either assemble the LOAD ADDITION routine yourself on a Monitor, or if you have saved it, LOAD"LOADER/1",8,1 and now use it to load the "GAME" on the disk. Surprise, surprise, the program crashes. Why? Load the program on this disk named "SYS10037" (the LOADER part of the filename is only for information, not part of the loadable filename), and disassemble it on your Monitor, (.D 2735). You will see that the following have been added.

```
JSR $FFD5 ; LOAD as before, but now JSR
STX $2D   ,stores LB of EA in location 45
STY $2E   ,stores HB of EA in location 46
JMP $A7AE ;warm start to start of Basic
(then the name in ASCII follows)
```

These were not needed in the all-MC program, but are needed if Basic is used. To run "SYS10037" merely enter just that. As by magic the disk program, "GAME" is loaded. Notice that JSR $FFD5 replaced JMP $FFD5. JSR is similar to GOSUB, and JMP to GOTO. So, JMP is used when it is at the end of a routine, and JSR when in the middle

AN AUTOBOOTING LOADER

If you want the loaded program also to RUN automatically, merely enter two new lines to the "SYS10037" program,

```
JSR $FFD5 ; as before
STX $2D   ; as before
STY $2E   ; as before
JSR $A659 ; new line
JSR $A871 ; new line
JMP $A7AE ; as before
```

(then the name in ASCII - and reset the SETNAM pointers as necessary).

Lo, your program loads and runs (But first save this as LOADER/2, with the appropriate EA). Let's see you do that with Basic!!! "SYS10037" may be located anywhere there is room for 34 bytes, it the SETNAM X and Y pointers are readjusted for the new name location. If the Basic program to be loaded is long this will, in fact, be necessary so that the loader is not overwritten whilst it is working. It can be placed in high Basic or at 49152 or 680, or even in the Cassette Buffer at 820 if you do not intend to use tape.
A similar routine could be used to load and run a MC program too, and this is particularly interesting because one could be loaded by a Basic one into an area not used by the Basic program. In that case;

```
JSR $FFD5 ;load as before
JMP $(BA) ;substitute the BA of the loaded MC prog.
(then the name in ASCII - reset SETNAM pointers as necessary)
```

In this case you could save the result as LOADER/3. So, /1 shows you how LOAD works in MC, /2 loads and runs a Basic target, and /3 loads and runs a MC one.

When LOADER/1 was attached to a Game Loader, the RUN routine was unnecessary, because the second file was self-running (as most are). On the disk you will also find the Basic progtramme, "SYS680 BOOT LDR". When you load and run it it puts a MC programme in memory at location 680, which includes the two autobooting lines for a Basic target. Decimal 680 is entirely separate from Basic (which starts at Decimal 2048), and so is not affected by it. If you load and run this Loader and then enter the command SYS680, it loads and runs the target programme, in this case, "GAMES". The Loader may be changed to load any programme you desire; follow the instructions given in the REMs.

If you will list this Loader you can then include those lines (without REMs or instructions, of course) in any other programme you please, allowing you to load and run any other programme from within it. This is remarkable: Say that Programme A has an option to SYS680. It it is chosen it loads and runs basic Programme B. It you include that routine (suitably altered for name) in Programme B it can then load and run Programme A again (or Programme C, etc)! The possibilities are limitless.

You can save SYS680 to disk from the SYS680 BOOT LDR by running the latter and then on your monitor,

```
.S"SYS680 BOOT",08,02A8,02DB
```

for a 16-letter name. This would have the advantage that it can then be loaded as MC from within a programme, to load and run another. This is much neater than using Dynamic Keyboard techniques, and does not clutter the screen with loading data. It is always a good idea to include the SYS number of any MC programme in the filename. It tells you automatically what command you need to run it!

DYNAMIC BOOTING

The MC solution is indeed a neater way to load and run a programme, but it does require the use of MC, which may daunt some people. On next months disk will be DYN KB BOOT which does it in Basic, clumsily but simply. Merely fill in the programme you want to boot, and include this at the end of your main programme.

FINAL THOUGHTS

Finally, all that we did here was to ask why the loader was loading the wrong (or no) file, and then to correct it by common sense and a little Machine Code. Some freezers make several copied files, and the ideas described here should work for each of them, if necessary. The result is very satisfying: the games now work with fast-loaders, so preserving sanity. And it is done by simple Machine Code; you do not have to be a prisoner of Basic, not of the gameswriters nor of indifferent copy programmes.

In the course of solving the loading problem, a nice MC routine was also developed for loading (and running) any programme

Is any reader dismayed that this article has talked openly about freezing copies? It is every owner's right to make a backup, and it is wrong for manufacturers to force you to buy expensive copies of their overprotected programmes. On the other hand, that protection is useless for anyone owning reasonably powerful copying equipment, and greatly inconveniences legitimate users. To give away copies of a copied programme is another matter; whether or not you sell them, it is piracy, read "theft".

In the present instance the original tape was bought. Piracy is as sensible as throwing out a book before reading it. If gameswriters cannot make a profit they will not write games for the Commodore, and then there will be no more programmes to copy. Have you killed any golden geese recently?

DESIGNING A ROLE PLAYING GAME

GORDON HAMLETT

To compliment our Adventure series, we start a new service for people more interested in Role Playing Games

Welcome to a new series of articles intended to fit nicely alongside the excellent ADVENTURE WRITING series from **JASON FINCH**. What I intend to do is examine the sorts of thought processes that you will have to go through in order to create your own role playing game. This will be done with reference to various commercially available games, looking at their strong and weak points.

What this series definitely will not cover is the programming aspect of games creation. I have great faith in the intellectual capacities of CDU readers as far as that department goes! Instead, I will present you with sufficient ideas for you to design your own stand alone modules or even a complete game.Who knows, you might even come up with something that be included on the disk in future issues.

One final word of warning. If you want to write a character generator program for your local Dungeons and Dragons game, that is fine. If you wish to publish such a program, you will first need to obtain various licensing agreements with the people who own the copyright. This will not come cheap so it might prove wiser to devise your own scheme of things. I will concentrate of fantasy themes simply because they allow the maximum scope for incorporating ideas Cowboys weren't renowned for their abilities as spell casters and combat in space tends to come down to variations on the theme of Zap! You're dead.

CHARACTER DESIGNING

This month's article will deal with designing a character. Anyone who has ever played an RPG will remember countless hours trying to create a party, rolling hundreds of dice and generally wasting a lot of time before the interesting adventures start.How much easier to get all the hard work done for you This module would be extremely useful either as an aid to real RPGers or as the start of computer moderated game.

The basic premise of every RPG is that each player has a number of characteristics which affect how well he performs in a variety of circumstances. To go back to D and D, the six characteristics are strength, intelligence, wisdom, constitution,dexterity and charisma.

CHARACTERISTICS

STRENGTH is self evident and no real alternative has been found. The stronger you are, the more you can carry and the more damage you do etc. **INTELLIGENCE** is usually needed for any magic users in the party but could also be linked to any problem solving ability required in the game. **WISDOM** is usually dropped altogether or lumped with intelligence. D and D used it for clerics praying to their various deities

DEXTERITY or agility is again an attribute common to all games. This will determine how well you can dodge blows, who reacts first in combat situations and what are your chances of falling off rickety old bridges. Dexterity is also vital to thieves when it comes to picking locks, dismantling traps etc

CONSTITUTION determines the potential health of a character and again, is common to all games in one guise or another. It is a measure of how fit and healthy a person is. Closely linked to constitution are hit points. These are a more dynamic measure ofa character's current state of health and the value of this variable will fluctuate constantly throughout the game as the player gets wounded and is healed etc. The usual rule of thumb is that when hit points reach zero, the player is dead. The final D and D trait is **CHARISMA** and this is one that usually gets ignored unless your game is a particularly involvedone. It deals with how other players and monsters in the gamereact when you try to talk with them and so on.

By and large, these values, by whatever names you call them,will remain fairly static throughout the game. When they do change though, they are likely to alter a whole range of variables that depend on them - chances of hitting an opponent,of being hit, damage caused etc. Possible reasons for altering aprime value might be some form of magic or paying for an appropriate training course.

OTHER FACTORS

There are several other factors that might affect these initial statistics. Many games allow characters to be different traces - dwarves, elves etc. Dwarves are strong and durable and so are likely to get bonuses for strength and constitution. On the other hand, they are not renowned for their intelligence and somay suffer in that department.

If you allow female characters, then they are likely to gain bonuses for constitution (women live longer) and dexterity but lose points for strength. And so it goes on. You are limited only by your imagination, desired degree of complexity and disk space!

Other variables that your character file might have to contain include some of the following. A luck factor (or saving throw) is used to see whether you avoided a trap or a spell etc. Age might be important. Size could be a factor too. Imagine if you are a giant trying to hit a hobbit. Surely there ought to be some penalty.

As characters progress through the game, so it is normal to award experience points to show how well they have done. This in turn leads to promotions, usually denoted as an increase in level. For example, a third level magic user would have more spells available than a second level magician and so on.

As far as combat goes, you will need to monitor the basic chance to hit an opponent and also a character's armour class -which determines how likely he is to be hit. These values are likely to be altered as a player acquires magic weapons and uses spells etc.

Characters with special abilities will need space to note the details. What spells a magic user knows or what are the chances of a thief performing different tasks are examples that spring to mind. You will also need to monitor what equipment is being carried and used. Finally, don't forget a name for your character. Players do like this sort of thing.

IT'S SIMPLE - REALLY

If all this sounds very complicated, let me try and simplify what you are trying to achieve. The character file will contain details of the basic values which in turn determine the out come of every event within the game. All the initial waffle is merely window dressing so that the player can set up the sort of character that he or she wants to take on the adventure. Once the initial values have been set, they can largely be ignored. Let the computer do the number crunching for you. If you want to seehow an expert system has been put together, I would recommend that you look at the Advanced Dungeons and Dragons rule books, particularly the Dungeon Master's Guide and Player's Handbook.

One final idea. It is very difficult to obtain a fair balance between characters that are too strong or too weak. Give everybody high values for their attributes and the game ceases to be a challenge. Make them too low and the character dies in his first battle. What you are looking for is an average plus character with room for improvement. You can become a super hero but it won't be easy. One possibility that as far as I know has not been covered yet is to have a built in character defect. Your best fighter has a thing about insects and will always run from battle with them. Someone else has a staff that always sings outwhen you are trying to sneak quietly past a guard and so on. In the next article, we will look at combat.

ADVENTURE HELPLINE

Jason Finch concludes this part of the helpline by giving you the final tips for finishing that great adventure by Tony Rome - KRON

First and foremost I must apologise for there being no Adventure Helpline last month. There are numerous reasons for that although if I start reciting them they will sound more like excuses. The one main mistake was that I didn't realise how close to the deadline date I was until it had nearly passed! I kneel down and beg for your forgiveness. But now that my grovelling is over, let's get down to the serious side. This is actually the last article in the Adventure Helpline series, relating to the adventure "KRON" that is! Over the past few months I have been looking privately at "The Astrodus Affair" published in the June 1990 issue and therefore next month I shall embark on the second topic of this series - that adventure. But first I must conclude the information relating to "KRON". There are sixteen locations discussed below, the first being the one that I left you with last time, and one of the others being repeated from a previous article. Once you have that information, finishing "KRON" should be child's play! So let's continue. .

40

In the Valley of the Dead. Due south over a deep lake stands the Castle of Spells, its stout walls rising towards the sky. Huge battlements overlook the lake and arched windows like cold dark eyes stare down into the valley.
Exits: EAST 39 (I think that is possible! - who cares if it isn't anyway?)
Type: PLAY FLUTE - the eagle swoops and takes you over to the battlements of the castle, location number 43.

OTHER INFO: IF YOU ATTEMPT TO NEGOTIATE THE LAKE BY GOING SOUTH THEN YOU WILL DIE INSTANTLY. THE LAKE IS DEEP AND ICY WHICH YOU JUST CAN'T COPE WITH!

41

At the northern end of a long passage. To the east is a room.
Exits: EAST 42, SOUTH 44

42

In the Alchemist's Chamber. A cauldron bubbles over a fire filling the air with a blue cloud. Sealed jars containing coloured liquids stand here.
Exits: WEST 41
Type: GET JAR

OTHER INFO: ONCE YOU HAVE GOT THE JAR LEAVE THE ROOM AS QUICKLY AS POSSIBLE OTHERWISE YOU MAY FIND YOURSELF FALLING ASLEEP. DO NOT OPEN THE JAR YET EITHER.

43

On the battlements of the Castle of Spells. A staircase leads downwards.
Exits: DOWN 44

OTHER INFO: SO YOU MADE IT OVER THE DEEP LAKE - NOW JUST ENTER THE CASTLE BY ENTERING DOWN.

44

At the foot of a stone staircase. There is a passage to the north and a door to the east
Exits: NORTH 41, EAST 45, CLIMB 43

OTHER INFO: WHEN ARRIVING HERE FOR THE FIRST TIME YOU SHOULD GO NORTH AND THEN FOLLOW THE CORRIDORS UNTIL YOU GET TO THE ALCHEMIST'S CHAMBER. THE JAR THAT YOU FIND THERE IS VERY IMPORTANT.

45

In a large hall lit with many candles. Doors are east, west and south. An oak chest stands in one corner.
Exits: EAST 46, SOUTH 47, WEST 44
Type: (Enter the following commands on entering the room once you have the princess, not the first time that you get here)
OPEN CHEST - you must have the keys because it is initally locked. You should find a knife.
GET KNIFE - this is vital
GET CANDLE - this is needed for when you go south.

OTHER INFO: AT SOME STAGE IN THIS ROOM, USUALLY WHEN YOU RETURN FROM HAVING FREED THE PRINCESS, A RATHER UNFRIENDLY CREATURE CALLED BALZAN WILL APPEAR AND TRY TO KILL YOU. IF YOU ARE TOLD THAT HE FIRES AT YOU, ENTER THE FOLLOWING IMMEDIATELY: RAISE SHIELD. THAT REFERS TO THE SHIELD GIVEN TO YOU BY THE BORAN MAN IN THE SECOND STAGE OF THE ADVENTURE. THE BEAM IS THEN REFLECTED BACK AT BALZAN AND HE SINKS TO THE FLOOR - DEAD.

46

In the Magic Room. The walls slowly move towards the centre of the room. There are doors to the north, east, south and west.
Exits: NORTH 50, EAST 50, SOUTH 48, WEST 45

OTHER INFO: THIS IS THE TIME WHEN "THE WALLS MAY MOVE YOUR SENSES SO". THE MESSAGE ON THE SCROLL TOLD YOU TO "TRY SOUTH, THEN

EAST, THEN SOUTH AGAIN", SO DO JUST THAT. IF HERE FOR THE FIRST TIME ENTER: SOUTH, EAST, SOUTH. OTHERWISE YOU WILL BE CONFRONTED BY THE OGRE OF THE CASTLE AND WILL BE KILLED INSTANTLY.

47

In a large empty room to the south of a hall. A window looks out to the lake far below.
Exits: NORTH 45
Type: LOOK - you see that you are in a mirrored room
EXAMINE MIRROR - you see a panel
PUSH PANEL - a staircase leading down is revealed
LOOK - you are presented with a new description
DOWN - to location number 53 The panel closes

OTHER INFO: NONE OF THE ABOVE WILL WORK UNLESS YOU ARE HOLDING A CANDLE WHEN YOU ARE IN THE ROOM. IF YOU ATTEMPT TO GO DOWN THE STAIRS WITHOUT THE CANDLE YOU WILL FALL AND BE KILLED. THAT WOULDN'T REALLY DO YOU MUCH GOOD.

48

In a passage Dimly lit passages lead in all directions
Exits: NORTH 46, EAST 49, SOUTH 50, WEST 50

OTHER INFO: IF ATTEMPTING TO REACH THE PRINCESS YOU SHOULD GO EAST AND IF RETURNING FROM FREEING HER YOU SHOULD GO NORTH, OTHERWISE YOU WILL BE KILLED.

49

In the Circular Room. Doors lead in all directions.
Exits NORTH 50, EAST 50, SOUTH 52, WEST 48

OTHER INFO: IF ATTEMPTING TO REACH THE PRINCESS YOU SHOULD GO SOUTH. IF YOU HAVE FREED HER YOU SHOULD GO WEST.

50

You are confronted by the Ogre of the Castle!... He kills you instantly...
Exits: none

OTHER INFO: UNLESS YOU WANT TO BE EATEN BY AN OGRE AND HAVE TO START ALL OVER AGAIN, I WOULD SUGGEST THAT YOU AVOID THIS LOCATION AT ALL COSTS!

51

Inside a small stone room. The heavy oak door reveals a dimly lit cell.
Exits: EAST 52
Type GET ZORA - just take her as if she's an object You have freed her and must now return to the Cave of Ice. See the information at the end regarding the "correct route".

52

In the Castle Dungeons A damp musty smell fills the air and water drips incessantly from the stone ceiling To the west is a small barred room with a strong oak door.
Exits: NORTH 49
Type: DROP JAR
NORTH - to 49
WEST - from 49 to 48
EAST - back to 49
SOUTH - back to 52
EXAMINE GUARD - you find some keys
GET KEYS
WEST - to location number 51

OTHER INFO: THE JAR CONTAINS A SLEEPING POTION AND SO WHEN YOU DROP IT THE POTION BECOMES ACTIVE. THE WHOLE IDEA IS TO SEND THE GUARD TO SLEEP AND NOT YOURSELF AND THEREFORE YOU MUST LOITER ELSEWHERE FOR A WHILE TO LET IT TAKE EFEECT. THAT IS THE PURPOSE OF THE FOUR DIRECTIONAL COMMANDS, THEY SIMPLY CREATE A TIME SPAN FOR THE POTION TO TAKE EFFECT. THEY ARE NEVERTHELESS ESSENTIAL. WHEN YOU HAVE THE PRINCESS, ENTER: NORTH, WEST, NORTH, WEST TO RETURN YOU TO THE LARGE HALL.

53

At the bottom of a long stone stairway. To the north is a silver door.
Exits: none
Type EXAMINE DOOR - only those that know the password can open it
OKURA - the door swings open. Just type that as it it were a verb.
NORTH - the candle burns out and you are returned to the Cave of Ice, location number 31, detailed last month but repeated below in case you missed it!

OTHER INFO: IF THE CANDLE GOES OUT THEN YOU WILL BE ATTACKED BY A BAT AND IT WILL KILL YOU. THERE ARE NO EXITS TO BEGIN WITH BECAUSE THE PANEL CLOSED BEHIND YOU, SEALING OFF THE STAIRCASE.

54

On a clifftop overlooking the Sea of Storms. Far below is a rocky inlet
Exits: none
Type RUB RING - you are returned to the place where you originally rubbed it unless this marks the end of your adventure.

OTHER INFO: THIS LOCATION COULD MARK THE END OF YOUR ADVENTURE IF YOU RUBBED YOUR MAGIC RING WHILST THE PRINCESS WAS "IN YOUR POSSESSION". YOU ARE THEN TOLD A LITTLE BIT OF

WHATEVER - I WON'T SPOIL THE ENDING FOR YOU, YOU SHOULD NOW BE ABLE TO FIND OUT WHAT HAPPENS FOR YOURSELF!!!

31

In the Cave of Ice. Ice and snow cover the roof and the walls. To the east over a narrow but deep pit is a passage. A silver door is set in the south wall. West is a tunnel.

Exits: WEST 30, JUMP 32

Type: EXAMINE ROOF - there is a pulley with a weight attached to some cord

CUT CORD - various things happen but more importantly a crown is revealed

GET CROWN

RUB RING - you have the princess, you have the crown, you should have your ring still, so rub your magic ring and whisk yourself away to the end of the adventure.

OTHER INFO: RETURNING HERE WITH THE PRINCESS MARKS ALMOST THE END OF YOUR ADVENTURE. YOU MUST HAVE THE CROWN BEFORE YOU CAN ESCAPE COMPLETELY AND THAT IS DONE AS DETAILED ABOVE.

FINALLY

That final chunk of information should allow you to finish the adventure. If you still can't then there isn't much I can do for you I'm afraid! As I have become accustomed to doing, I shall now reveal one of the "correct" routes through this section of the adventure. The location that you should visit are in the order as follows: 40-43-44-41-42-41-44-45-46-48-49-52- (49-48-49-52-) 51-52-49-48-46-45-47-53-31-54

The important thing to remember in this section is never to enter location number 50 - it means certain death. Also remember what to do when you drop the jar and then, later, are confronted by Balzan. Over the past six articles I have shown you in increasing detail how to complete "KRON". Hopefully you will have sorted out any problems that you had with it and will now be able to successfully complete it. If you have any further queries about "KRON" then please write to me at: Adventure Helpline (KRON), 11 Cook Close, Brownsover, Rugby, Warwickshire, CV21 1NG. Please note that I shall only be dealing with requests regarding "KRON". See you all again next month, hopefully, when I shall begin to pull back the curtains to reveal the method for completing yet another of CDU's excellent adventures. Unfortunately there are no AW programs on this months disk as promised last issue. They will now be on the January disk. My apologies.

WORTH THE HASSLE??

We continue the investigation started last issue, into the rights of consumers when treated unfairly by Companies

JASON FINCH

This article is continued from last month and just in case any of you missed the start, this is an interview between solicitor Jonathan Lines and myself on the subject of unreasonable companies and what the customer can and cannot expect to receive. As with last month's article the text in bold represents myself and the other is the solicitor's reply. So without using up further space, let the interview continue...

LOSS OF EARNINGS

SAY THAT IF SOMETHING IS UNDER GUARANTEE AND YOU SEND IT BACK TO THEM, THE COMPANY HAVING TOLD YOU THEY WILL BE ABLE TO SEND ANOTHER OUT WITHIN A WEEK, AND IT TAKES A MUCH LONGER TIME - SAY THREE TO FOUR MONTHS - AND YOU HAVE LOST WORK IN THE MEANTIME BECAUSE YOUR JOB DEPENDS UPON THAT PIECE OF HARDWARE, SAY IF YOU WORK FROM HOME AND PRODUCE ADDRESSED LABELS FOR A COMPANY, ARE YOU ALLOWED TO CLAIM COMPENSATION FOR WORK THAT YOU HAVE BEEN UNABLE TO DO BECAUSE THE COMPANY TOOK A CONSIDERABLY LONGER TIME THAN THAT ORIGINALLY STATED?

Not as a rule, I would have thought, because if they fail to perform a guarantee, fail to perform a contractual obligation, in a reasonable time, then if they do have any liability arising out of the that, it's got to be reasonably foreseeable I think and it's all a question of the facts there. But there is no principle that simply says that they should have realised that if they didn't return the goods to you, you were losing x-hundred pounds worth of business in the meantime. You may have been, but it doesn;t necessarily follow that that can be claimed directly back from them. There may be a case in suitable circumstances under that sort of heading, but I think that is more likely to be the exception rather than the rule. I think that that would be the exceptional case where it would be obvious to them that you would lose something if they kept it longer than a "reasonable time". Legally, you are obliged to keep your losses to a minimum. If there are ways and means reasonably available to you to use other equipment in the meantime, then if you choose not to do that you can't claim any losses whatsoever. So it has very severe limitations that one. I don't wish to say that there will never be a claim like that - but if you have a claim, it is likely to be smaller than you think, and there is a good chance that you will end up with no claim at all

THE WRONG PRICE

IF THE CUSTOMER ORDERS A PRODUCT THAT COSTS, SAY TWO HUNDRED AND FIFTY POUNDS, AND IN THE PROCESSING OF THAT ORDER THE COMPANY MAKES A MISTAKE AND SENDS OUT A MODEL THAT COSTS ONLY TWO HUNDRED POUNDS, IS THE CUSTOMER ENTITLED TO THE CORRECT ONE WITHOUT PAYING THE ADDITIONAL FIFTY POUNDS?

No, I don't see why that should be. You would then have the option of accepting the two hundred pound model and paying two hundred pounds for it or them correcting the mistake and you paying the extra for the correct model. There is no reason for you not to pay that extra cost if they later provide the correct model. It's like it you go into a shop, having seen something in the shop window that is incorrectly priced. They then say "oh you can't have that for twenty pounds - it should be thirty." They are perfectly entitled to say that. They are not obliged to sell it to you for the figure they have advetised it in the shop. That's not quite the same point. But there is no way you could say you're entitled to the correct one for only, in this case, two hundred pounds. When you are considering these points, it is always good to consider the most extreme situations, no matter how ridiculous they may seem. If you order something that costs two and a half thousand pounds and then purely because of a mistake in the processing of the order, perhaps they missed off a digit, you are sent something that costs only two hundred pounds, you are clearly not entitled to say "well jolly good, I'll have the correct item for two hundred pounds, please!" In that sort of situation I think the answer is fairly clear.

WHO TO DEAL WITH

THERE IS A MAIN COMPANY THAT SUPPLIES YOU WITH THE GOODS - THE MAIN COMPANY SUPPLIES THE CUSTOMER - AND THEN ABOVE THEM YOU HAVE THE LARGER COMPANY THAT SUPPLIES THEM. NOW IF THE MAIN COMPANY DOES NOT HAVE WHAT YOU WANT IN STOCK, AND YOU PAY THEM AND THEY DEAL WITH THE ORDER, BUT THEY GET THEIR SUPPLIER TO SEND IT DIRECT TO YOU. AND THEN IF THOSE GOODS ARE FAULTY OR WHATEVER, WHO DO YOU DEAL WITH? IS IT THE COMPANY THAT YOU PAID OR THE PRIMARY SUPPLIER?

Basically the company you paid - they are the ones that you have the contractual arrangement with. So they are the company that you continue to deal with. There are circumstances when you can be involved with the original supplier direct but that is usually under a guarantee or something, or under a particular circumstance arising out of the equipment, for instance if it exploded and caused you physical injury, then you would sue the manufacturer on that direct - but that is a different story and obviously not connected with what we are talking about here.

SO JUST BECAUSE THEY SENT IT TO YOU DIRECTLY, DOESN'T MEAN....

....doesn't mean that you go directly back to them if there is anything wrong with it. You should continue dealing with the people that you paid for it.

IS THERE ANY OTHER INFORMATION THAT YOU THINK MAY BE USEFUL TO CUSTOMERS - THEY DON'T REALLY HAVE THAT MANY RIGHTS DO THEY?

Well, yes, but I tend to play down the rights a bit because they may exist and you always ought to uphold them and claim them but one tends to assume, certainly in my job, that the question only arises because a company is being unreasonable. But then we come to the real terms of how important your rights are to you in terms of spending money and pursuing them. Unless there is a sufficient amount of money involved, the situation is usually to accept what is offered by the company. Sueing in courts has its expenses and one tends not to go that far unless the case is sufficiently blatant or there is a sufficiently large amount of money involved. But the rights are there in terms of supplying goods that are adequate for the purpose and so on. One other danger that I may mention in relation to wanting money back or claiming a refund and having goods replaced, there is the question that if you use the goods there comes a point, probably quite early on, a time when it can be said that you have accepted those goods and you would then lose any right to a refund even though they are not quite the goods that you ordered. If you are going to claim then get on with it So often claims are lost becuase people just hesitate and don't pursue their rights. So the rights there - the problem comes when the company doesn't accept them. A good responsible company usually will provide a good service. A good thing to do before dealing with a company is perhaps check with other customers, if you are able, as to the sort of service they were given and whether that company has a good after-sale service - whether it regards that as important or not. If it does then you are not going to have 90% of these problems because the company will be co-operative if any problems arise. The trick is to be able to deal with a company that has a good attitude. If you are dealing with a company that doesn't want to know, then whatever your rights are, you will have any uphill battle.

IN CONCLUSION

From my interview with Mr. Jonathan Lines you may decide that really there is not much that you, as a customer, can really do if a company is being particularly awkward. But you must remember that the over-riding idea that I found made itself apparent was that it's one thing to have rights but you must realise the limitations of those rights if you are dealing with someone who is being very awkward. From my own experience I would like to add that you must always be very persistent and never lighten up for the slightest moment.

When I, recently, wished to send some faulty goods back I was told that I must pay the postage As I knew, and Mr. Lines backed me up, there was nothing to say that I should not do so. But the minimal cost of four pounds only covered Parcel Post which I was told can take a week to reach its destination. There are people who don't want to, or like me can't, wait around for an eternity whilst parcels get on their way and so forth. Be persistent and if there is an ounce of understanding in the company they will assist eventually if you keep pushing. It worked for me anyway. Go to a higher authority within that company if you can and don't wait hours if you are told that someone will ring you back. If you were told that you would be called back soon and haven't been, you call the company again yourself. It may cost a bit in telephone bills but I believe it gets results!

Of course, with the really awkward companies you can go a stage further, as Mr. Lines detailed in his first answer referencing your local Trading Standards Office Although everything would seem to point to the fact that you must let the company take the lead in sorting out your problems, you can always hurry them along. But if it is quite obvious to you that the company, although slow, is doing everything that it can, don't apply the pressure too much or you will find that a role-reversal occurs and you become the irritation! There is a fine line between being persistent and positively irritating. One thing that I would advise is that before you do anything extreme, like taking a company to court, think rationally and apply cold logic - don't just jump in with the idea "well I've got a principle and I'll stick to it come hell or high water!". That sort of thinking can do you no good sometimes Everything should be weighed up and you must decide what is and what just simply isn't "worth the hassle" of bothering with.

As a definite end, if a company is being unreasonable, simply be persistent, don't accept any excuses, and more importantly, don't back down for a single moment. If talking and complaining over the telephone gets you nowhere then write a letter to the manager or managing director of the company. That, I'm afraid, is all the adivce I can offer you. I hope that you have found this two part article both helpful and interesting and I sincerely wish you good luck should you ever be confronted by a truly awkward company - remeber, try and get a recommendation from someone else first.

MACHINE LANGUAGE GEMS

One or two readers are at last sending in their hints and tips for others to benefit. We present five this month, hopefully these will be the first of many BONES

1) MESSAGES

During the course of a game or utility which you might be creating, it is often necessary to print various messages or whatever to an area, or window, of the screen.

This small routine can take care of delivering the messages for you. Each message must be preceeded with a byte containing the value of 64 (or "@"), and the final message in the message list must finish with a byte of 64 ("@").

Data example of two messages.

```
100 ; *** first message ***
110 BYT "@This is a typical",255
120 BYT "message",255,255
130 BYT "with a blank line above this one"
140 ;
150 ; *** second message ***
160 BYT "@This is message number two",255
170 BYT "and the last message"
180 BYT "@"
```

Where you place your equates within the source code you can enter a screen location value for the 1st position to print to within your message window, or area etc. ie; SCRNPOS EQU 1270. You could, if you wished, change this to a double byte variable which could then be changed to service different message boxes, or whatever

Messages using this routine are limited to 254 bytes in length, this includes line seperators (255). There is not much call for messages longer than this. However, with a little thought, this could be fairly easily changed to accomodate much longer messages. No account of colour has been established, but once again, with a little thought each message could have a corresponding block of colour data used for each character of the message string

Because this routine pokes characters directly to screen locations, ASCII control codes cannot be used, although you are able to use the ASCII character codes to place in your message data if your assembler accepts these (most do). The program automatically converts the ASCII into CBM screen codes.

On the disk is a small 'bare bones' demo program of the routine at work. It is located from $C000 to $C425. The breakdown of the code is as follows.

LABEL	LOCATIONS
MAIN	C008 - C03A
PRINTMESS	C03D - C08C
DELAY	C08E - C0C4
SFX	C0C5 - C0CF
SETUP	C0D0 - C0F1
WINDOW DATA	C0F2 - C1CF
MESSNO	C1D0
CNT	C1D1 - C1D2
MESSAGE	
0	C1DD - C242
1	C243 - C27F
2	C280 - C2E9
3	C2E4 - C334
4	C335 - C3AC
5	C3AD - C425 (blank spaces to clear message area)

2) SINGLE CHARACTER ADDRESS

Here is a useful routine to locate the HI/LO address of a character from within a character set.

```
01    CHARSET = $2000 ; or wherever the set is in
        memory
02      LDA #>CHARSET ; HI byte of charset start
        address
03      STA $FC
04      LDA CHARVAL
05      ASL A
06      ROL $FC
07      ASL A
08      ROL $FC
09      ASL A
10      ROL $FC
11      STA $FB
12      RTS
13      CHARVAL   BYT 0 ; store the character you are
        ; working on here.
```

On return from the routine the HI/LO address is stored in $FC/$FB respectively.

3) SPRITE SCREEN ADDRESS

This short routine will calculate the screen address of sprite X/Y coordinates Can be used, for example, with a sprite pointer. The routine as presented, specifically calculates for sprite 0 with the screen in bank 0.

```
01CONVERT
02              JSR FINDX
03              JSR FINDY
04              LDA #4
05              STA $FC
06              LDA #0
07              ADC CHARX
08              LDY CHARY
09              BEQ EXIT
10 LOOP         CLC
11              ADC #40
12              BCC SKIP
13              INC $FC
14 SKIP         DEY
15              BNE LOOP
16 EXIT         STA $FB
17              RTS
18 FINDX        LDA #29
19              STA TMP
20              LDA $D000
21              PHA
22              LDA $D010
23              AND #1
24              BNE SKIP2
25              PLA
26              SEC
27              SBC #24
28              PHA
29              LDA #0
30              STA TMP
31 SKIP2        PLA
32              LSR A
33              LSR A
34              LSR A
35              CLC
36              ADC TMP
37              STA CHARX
38              RTS
39 FINDY        LDA $D001
40              SEC
41              SBC #50
42              LSR A
43              LSR A
44              LSR A
45              STA CHARY
46              RTS
47 TMP          BYT 0
48 CHARX        BYT 0
49 CHARY        BYT 0
50 END
```

On return from CONVERT, $FB/$FC hold LO/HI bytes of screen location (1024 - 2-23). CHARX contains screen X coordinate (0-39). CHARY contains screen Y coordinate (0-24)

4) MULTIPLICATION

Multiplying two numbers can prove a little tricky in machine language, so here is a routine which may assist you.

```
01              P1 = $FB
02              P2 = $FD
03 MULTIPLY     LDA $0
04              STA P2+1
05              LDX #8
06 SHIFT        ASL A
07              ROL P2+1
08              ASL P1+1
09              BCC CHCNT
10              CLC
11              ADC P1
12              BCC CHCNT
13              INC P2+1
14 CHCNT        DEX
15              BNE SHIFT
16              STA P2
17              RTS
18              END
```

USAGE: STORE MULTIPLICAND IN P1
STORE MULTIPLIER IN P1+1
Product result in P2/P2+1

5) DESIGN STRUCTURE/ ALGORITHM AND FLOW CHARTING

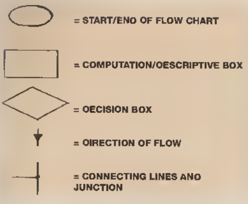
The first solution to a problem is rarely the best solution as we shall now find out.

ALGORITHM STEPS

1. **Define the Objective**
2. **Break the objective down into sub-problems**

Our objective, in this exercise, is to create a small routine which will take care of an animated sprite explosion sequence. We shall assume that there will be eight frames with which to depict the explosion, and with the frame image pointers being from 160 through to 167 consecutively. We shall also assume sprite number zero to be the sprite we are going to use with which to display the explosion. More usually the explosion would use the sprite which holds the image of that which is being exploded...for example an alien, or energy bolt. This being the case, there is no need to turn on the sprite itself as this would already be turned on as either an alien, bolt, or whatever other image.

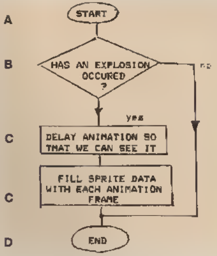
Before we start there are the five main diagrams which I use to create flow charts. (These are very much standard).


= START/END OF FLOW CHART
= COMPUTATION/DESCRIPTIVE BOX
= DECISION BOX
= DIRECTION OF FLOW
= CONNECTING LINES AND JUNCTION

So, we have defined the objective - to create an animated sprite explosion - We now need to break this down into an algorithm of sub-problems.

1. Start
2. Test for an explosion
3. If yes then display each animation frame with a short delay so we can see it.
4. End

Having done this we can now outline a first flow-chart.


A
START
B
HAS AN EXPLOSION OCCURED ?
no
yes
C
DELAY ANIMATION SO THAT WE CAN SEE IT
C
FILL SPRITE DATA WITH EACH ANIMATION FRAME
D
END

This is the initial set of requirements to create the explosion.

In (A), a variable will be required to represent a true or false situation. If the variable is true, ie; =1, then continue with the explosion situation, but if false (0) then return to the caller. The caller being that part of the main program which calls the explosion routine - usually tied into a main-looping program section.

We also need (B) to delay the frame presentation of the animation in order to see it execute, otherwise it would be too quick, most especially in machine code. The length of the delay will eventually be 'finely tune' once the routine is up and running correctly.

(C) will be the guts of the animation sequence, delivering each frame image to a sprite until all frames have been displayed.

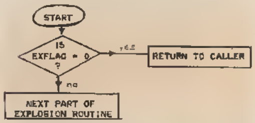
Finally (D). Here we need to close the routine by resetting the various variables and turning off the sprite.

SMALLER CHUNKS

Now we need to look more closely at the initial algorithm outline and break it down a little further.

(A) During the program SET-UP we will need to initialise a variable called EXFLAG (EXplosion FLAG) to equal false, ie, 0. Thus, EXFLAG = 0.

The first line of the explosion routine will simply be a check on the state of this flag. If an explosion is required we simply set EXFLAG = 1. This would most likely occur during a collision test which proves positive.



Please note that although I have used a full byte for the YES/NO, TRUE/FALSE, variable EXFLAG, a single BIT in a byte could also be used where eight seperate truth flags could be incorporated.

(B) The animation delay requires a little thought - we cannot simply place a delay loop in here, otherwise the entire program would slow down whilst the explosion is being animated.

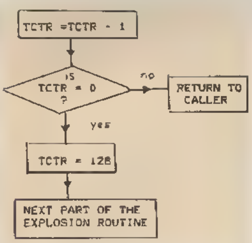
Rather than this, we need to use another variable, which we can call TCTR (Timer CounTeR), and preset this during program set-up to equal 1 (TCTR = 1).

When EXFLAG is set equal to 1, as I said earlier, probably during a positive collision test, we decrement the TCTR variable and then test to see if it has reached zero.

1. Reached zero, so reset TCTR to 128 (I have used 128 as an arbitrary delay value, fine tuning will come later), then move on to (C). The reason TCTR is set to 1 at SET-UP is to ensure that there is no delay on the the first iteration of the explosion routine.

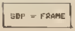
2. Not reached zero, so do not execute anymore of

the routine but return to caller so the rest of the program can get on with whatever it needs to do next.
As you can see, the use of a counter such as this will only execute the animation each time TCTR reaches zero, thus causing an animation delay without impinging upon the rest of the program.
Therefore:

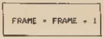


(C) One or two things have to be taken care of within this section, which is the main 'guts' of the explosion routine.
1. Transfer the frame image to the sprite data pointer.
2. Increment the frame ready for the next image.
3 Test to find the end of the animation sequence.
3a. If found not true then return to caller.
3b. If found true then pass control to (D)

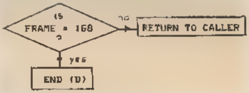
As program SET-UP we would need to set a constant, SDP (Sprite Data Pointer) to equal the start memory location of the eight sprite pointers, ie; 2040 (assuming bank 0), SDP = 2040. We also need a variable to contain the value of each frame image data to be placed into SDP - the first frame being equal to 160. So, FRAME = 160. Therefore,
1 would simply be;



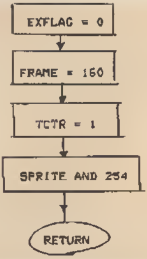
2 would get the variable FRAME ready for the next image;



3 because we increment FRAME variable after displaying the previous image, we must check FRAME for the final image plus one to determine if the end of the animation sequence has been reached. At the begining of this exercise we nominated eight animation frames consecutively from 160 onwards, so therefore, the value of 168 in FRAME would indicate that the last frame is



being displayed.
(D) The sequence has finished, so the routine must now close itself by setting the variable EXFLAG to zero, ready for the next explosion sequence, and to prevent the current one from continuing and displaying frames which had nothing to do with the explosion. Also it must set FRAME back to 160, the start frame for the next occurance of explosion. It also needs to set TCTR to a 1, to ensure no delay for the first frame, or iteration, and finally it must turn off the sprite.
So, the final part of code is;



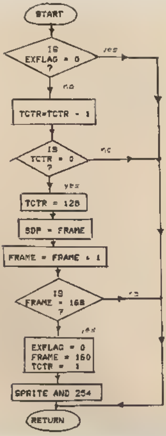
We have now extended our original algorithm thus;
1. Test for explosion flag truth..EXFLAG = yes/no
2. If yes then continue, else return
3. Decrement the delay counter TCTR
4 If zero then continue, else return

5. Set TCTR to the delay between frames length, initially set to 128 (see note below)
6. Transfer frame image (FRAME) to sprite data pointer (SDP)
7. Increment FRAME
8. If FRAME <> final frame+1 (168) return
9. Reset variables, EXFLAG=0, TCTR=1, FRAME=160
10. Turn off sprite
11. Return

NOTE: Once the routine is up and running, adjusting this value (128) will slow down or speed up the animation display. Adjust this value until found suitable. Increase to slow down. Decrease to speed up.

We can now put each section of the algorithm together to form a flow chart for the complete routine.

SET-UP-EXFLAG = 0 :TCTR= 1: FRAME = 160 :SDP = 2040 :V = 53248 (sprite base address)



Checking through the flow chart indicates that each section should work and take place at the appropriate time, so now we can convert this into working code.

BASIC(at SET-UP)

```
10 EXFLAG=0:TCTR=1:SDP=2040
15 FRAME=160:V=53248
   (subroutine explosion)
100 IF EXFLAG THEN 120
110 RETURN
120 TCTR=TCTR-1
130 IF TCTR THEN RETURN
140 TCTR = 128
150 SDP = FRAME
160 FRAME = FRAME+1
170 IF FRAME <>168 THEN 190
180 EXFLAG=0:FRAME=160:TCTR=1:POKEV+21,PEEK
    (V+21)AND(255-1)
190 RETURN
```

ASSEMBLER

```
10              EQUATES
12 SDP          EQU 2040
13 V            EQU 53248
14 ;
15 EXPLOSION    LDA EXFLAG
16              BEQ SKIP
17              DEC TCTR
18              BNE SKIP
19              LDA #128
20              STA TCTR
21              LDA FRAME
22              STA SDP
23              INC FRAME
24              LDA FRAME
25              CMP #168
26              BNE SKIP
27              LDX #0
28              STX EXFLAG
29              INX
30              STX TCTR
31              LDA #160
32              STA FRAME
33              LDA V+21
34              AND #254
35              STA V+21
36 SKIP         RTS
37 EXFLAG       BYT 0
38 FRAME        BYT 160
39 TCTR         BYT 1
40              END
```

I hope that this small demonstration of MACHINE CODE GMES, (Messages, Design Structure, Algorithms and Flow Charting), has helped in some small way to discover your own talents as a programmer. Maybe you too will soon submit your own utilities, examples or tips to CDU for evaluation with a view to publication, and fame

COLOUR TABLE EDITOR

NOT JUST ANOTHER GRAPHICS UTILITY

SIMON COLLINS

To compliment last months RASTER BAR EDITOR we give you another graphics utility for colour tables

The colour table editor was written by me for a specific purpose - I wanted to design a huge colour table for an intro sequence. Then I wondered whether the readers of CDU would be in the same position - not wanting to re-assemble the same program hundreds of times merely to test a colour table.
So I wrote this program, bearing in mind that people may want varying widths of raster line, and varying sizes of table.

LOAD AND GO

The finished program allows the width of the raster bar (the BAR SIZE) to be altered easily (keys B and Shift/B.) The length of the table can be altered as well (using the keys T and SHIFT/T.) The table will also scroll upwards when requested. This feature can be turned on and off using the S key.

During editing, the table will scroll to the position in the table that the cursor currently resides at. However, some may find this a hindrance. In this case, locate the cursor in a place in the table you wish to view, and press L. Once you do this, the table locks in that position until you press L again.

To change any value in the table, you simply use the plus (+) and minus (-) keys. Note that Shift/+ is the same as -, and that Shift/- is the same as +. This is due to the method that they are programmed - and also useful for people who only want to use one key (like some!)

DIFFERENT MODES

There are two modes for viewing the full, or edit, table at the top of the screen. The first is the default mode, called NUM, and displays the colours as numbers. The second is called COL, and displays the edit table as blocks of colour. In the first mode, the cursor is white, and in the second mode, the cursor is a star. You can swap between these two modes with the V key.

If ever you want to clear out all the table, simply change the current value in the table to the colour to clear the table to, and then press CTRL and C.

CTRL and T is the final option I shoudl mention - the value under the cursor becomes that last value accessed in the table when CTRL/T is pressed.

H will take you to the help page, as I explained, but F1 (the grey function key) will take you to the tiles mode, where drive #8 can be used to store tables for later use, and to use DOS commands, display a directory, and so on. All the keys are displayed on the screen at this point, so I won't go into them here.

Well, I could take forever explaining how this thing works, and still not finish, so if there's anything you don't understand, take my advice - TRY IT OUT. Then watch what it does.

You can load COLOUR EDITOR from the menu, or by typing LOAD "COLOUR EDITOR" ,8 and then typing RUN when the "READY." prompt is displayed.

MULTI TASKING ON THE C128

We recreate our series, started December 1989, on implementing a multitasking system on the C128

DAVID KELSEY

12 MONTHS AGO WE STARTED A SERIES ON MULTITASKING CAPABILITIES ON THE C128. DUE TO TECHNICAL PROBLEMS WE HAD TO POSTPONE THE SERIES. WE ARE GLAD TO SAY WE CAN NOW CONTINUE. TO REFRESH YOUR MEMORY WE HAVE REPRODUCED THE FIRST PART OF THE SERIES AGAIN.

Multitasking was a term that was associated with mainframes and minicomputers, but has only recently been associated with micro computers with machines such as the Amiga providing this feature. So what is multitasking ?

Multitasking is the term used to describe a computer which 'appears' to be doing several things at once. You will have experienced a form of multitasking on your commodore already. I'm sure that in some time or another, you have played an arcade game on your commodore. There you have objects moving on the screen and music playing in the background. Well the computer is appearing to be doing several things at once. Therefore by the above definition this is multitasking.

ABOUT THIS ARTICLE

The purpose of this article is to introduce multitasking in a general sense and to take the reader through the development of a very simple operating system that provides multitasking on the commodore 128. It will also introduce techniques for designing programs to run in a multitasking environment and to be able to design extensions to the operating system thus making it more powerful. This article is meant for people with an understanding of machine code and some understanding of how the commodore 128 works although I will endevour to explain the concepts to the best of my ability.

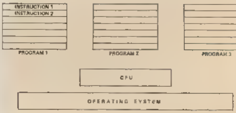
MULTITASKING OPERATING SYSTEMS

A multitasking operating system allows a user to load and run many different programs into the same machine and have them all running at the same time. But how is this possible on a commodore 128. There is only one processor (ignoring the Z80 as this cannot be used as the same time as the 8502 and this system is designed purely for the 8502.) so only one instruction from any one program can be executed.

The answer lies in a clever bit of software which 'switches' which program is being run on the micro-processor.

FIG 1

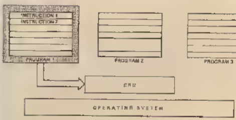
In FIG 1 we have several programs and one CPU. what we want is for all these programs to appear to be being



processed by the CPU. What the operating system does is say: let the CPU see program 1 and start executing it.

FIG 2

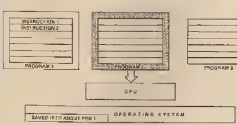
After a certain amount of time, The operating system gets



control and says:-
right that is enough of that program, saves all relevent information about where the CPU got to while running that program, and move onto program 2. ie let the CPU now see program 2 and start executing it.

FIG 3

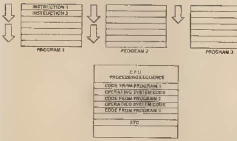
This is repeated for all the programs it can see. When it has gone through all the programs once, the operating



systems points the CPU back to program 1. It also restores the information saved about that program and allows the CPU to continue executing program 1. As the CPU starts back on program 1 where it left off it just appears as though nothing has happened except a slight delay. Therefore the sequence in time for the programs and the CPU are shown above.

FIG 4

A certain amount of time is allocated for each program. When that time limit has expired, then the operating system stops the CPU executing one program and points



the CPU to another program to run.

IMPLEMENTING THIS ON C128

The first question is how can this be done on a C128? My first thought was if it was possible to have BASIC programs appear to run simultaneously? I decided that it was possible but because BASIC on C128 uses a lot of memory locations to keep track of one program, when we switched to another program all this information would have to be saved so that when the BASIC intepreter returned to that program it could resume as though nothing had happened. The amount of memory that would have to be saved would mean that a noticeable delay would result while the information was saved. This would not be satifactory and so I decided that it was impractical. However in C64 mode, not nearly the same amount of information has to be saved so the concept is far more practical. In fact I recently ready an article in "Compute's Gazette" describing such a system as well as providing the software to do it. My approach deals with the multi-tasking of machine code programs. Before even starting I had to decide on a method which determined when to stop running one program and start another. 2 options were available :-

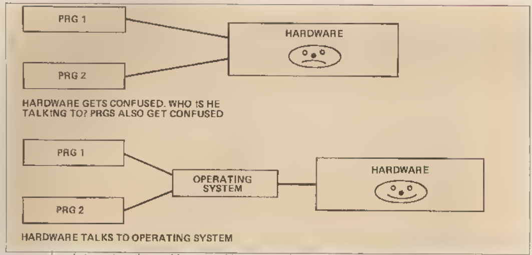
1. BY INTERPRETING EVERY INSTRUCTION OF THE PROGRAM

This method means that after a certain number of instructions The operating system could save the information and start intepreting another program in memory. The good bit about this would be that I could design my own language, but the code for a new language would be quite long. I could just interpret a machine code program. This would allow control over what the person tried to do in his code and could stop the program from doing anything potentially dangerous (this will be explained later in the article.). Also a tracing facility could be built in and information about a program running could be kept.

2. USE INTERRUPTS AND ALLOW THE CPU TO HAVE DIRECT ACCESS TO THE CODE.

This method is far more efficient allowing code to run directly on the CPU means that programs will run much faster than method 1. Also no interpreting code would have to be designed. The only problem is that, because a machine-code program could do anything, a potential time-bomb could be written which could cause the operating system to crash. Again the reason for this will be explained later along with a solution.

The decision was made to go for method 2, providing the most efficient system and provide guidelines to stop


PRG 1
PRG 2
HARDWARE
HARDWARE GETS CONFUSED. WHO IS HE TALKING TO? PRGS ALSO GET CONFUSED
PRG 1
PRG 2
OPERATING SYSTEM
HARDWARE
HARDWARE TALKS TO OPERATING SYSTEM

programs from being written that could cause problems.

THE FIRST STEP

Ok, An operating system will be designed that can allow several machine code programs to be run at the same time. I want a minimum amount of restriction on these programs so several questions are raised

1. How can several programs be run at the same time without at some point using the same memory location or creating havoc with the system stack ?
2. What if the program starts playing with the configuration register and trying to call KERNAL routines ?
3. What if one program wants to use interrupts which are also used by the operating system ?

Part of the answer to question 1 was to have for each program, it's own copy of page 0 and page 1 As these pages are special these were the only ones considered to have individual copies for each program. To stop programs from using other locations a guideline should be enforced to say that within your machine code program all memory locations used should be contained within the start and the end of a program Only the memory locations within the program block can be used by that program. There may be instances where a program will have to use memory locations outside of its program block (for example writing to the screen) but this will all be covered later

The Operating system should provide all routines required to control Hardware. This has to be done because several programs running at the same time may want to use the same bit of hardware. Obviously this cannot be done and some sort control has to be provided to share out the resource properly.

FIG 5

As the kernal routines for the original operating system weren't designed for a multitasking evironment, they should never be accessed. Memory is also hardware and so will be controlled by the operating system. Thus there will never be a need to use the configuration registers ($ff00-$ff03) and so should never be used.

Question 3 is difficult, and the easy solution is just to say interrupts can never be used. See the section on expansion later in this article for further ideas.

MEMORY

Right, this is the first consideration of the operating system and some fundemental ideas have to be explained before any progression can be made. Imagine several programs have been written. They have been designed at specific areas in memory (not necessarily for any reason) and we want all these programs to run at the same time within the multitasking evironment. The problem is that program areas will overlap and so if they were just loaded in at their designed locations, programs would lose part of their code as one program overwrote another either when it was loaded or while another program was running. This situation could cause the whole operating system to crash and obviously needs to be resolved.

The solution is to 1st keep track of what memory is available and what memory has been used. This means that all used memory locations apart from page zero and page one must be contained in the program block so that all memory used can be known to the operating system. The second part of the solution is to provide a program relocator that takes a program in memory and changes the addresses used so that it can be used properly ie it can run at the address the operating system has decided to place it at. For example consider this very simple program.

```
$3000 LDY #0
$3002 LDA $3200,Y
$3005 STA $3400,Y
$3008 INY
$3009 CPY #$50
$300B BEQ $3010
$300D JMP $3002
$3010 BRK
```

The program start = $3000
end = $34FF
(this includes all the code and data used by the program)

Now suppose memory wasn't available at location $3000 but there was enough memory at $4000 this program could be loaded at $4000, but obviously it couldn't be run because the addresses are wrong. It would work if the addresses were changed as follows :-

```
$4000 LDY #0
$4002 LDA $4200,Y
$4005 STA $4400,Y
$4008 INY
$4009 CPY #$50
$400B BEQ $4010
$400D JMP $4002
$4010 BRK
```

This would be the job of the relocator. It would make a program executable in memory.

TALKING WITH THE OPERATING SYSTEM

The user will have to be able to talk to the operating system in order to load in programs and run them. This means that some user interface is required. So a facility has to be provided to accept and process commands.

SWOPPING PROGRAMS

This Is the most important part of the whole system. When a program reaches the end of it's time slot and the next program is to continue it's execution, what information has to be saved ?
The obvious ones are the 8502 registers. These are A,X,Y the stack pointer and the status register I also mentioned that each program has his own page 0 and page 1, so these might have to be saved. In the real case however this doesn't have to be done. One of the features on the MMU is that the you can specify where in memory page 0 and page 1 are. All that has to be done is say that for any particular program. 2 pages are reserved for page 0 and page 1 and all that is required is that the MMU points to them and that the memory is flagged as used by the operating system. Therefore each program can have it's own page 0 and page 1 as well as the operating system. You don't have to move data In an out of locations $0000-$01ff. all that has to be done is the moving of pointers in the MMU.
When an interrupt occurs, all the information about the state of the CPU is stored on the stack. If the stack is moved to point to a new stack, all the information about the program which was being executed by the CPU is effectively stored. The only thing thats needs to be recorded is the STACK POINTER. This has to be done as we have to know where in the stack the data is. There is only one stack register that has to be shared by several programs.

THE STORY SO FAR

The operating system that will be designed will use Interrupts to swop programs.The interrupts to be used will be IRQ Interrupts. This allows for some interesting programming techniques which will be explained later.

FURTHER CONSIDERATIONS

What happens if a BRK command is encountered when a program is executing. This has to be dealt with and the simple solution is just to remove the program from memory. (similar to QUIESCING it. See later).

Storage will have to be monitored. The operating system will need to know what memory is available and what memory has been used.

A program table is also needed. The operating system needs to know some information about a program so that it can switch to different programs. Information such as program name, where the program's page 0 and page 1 can be found and where the program Is actually located in memory.

Some method of program relocation will be required as explained in the last part of this article. This would be required once the program has been loaded into memory.

Some sort of Interface is required for the user to communicate with the operating system. How can this be done ? The way I decided was to have a key-press that would signal the operating system and thus prompt me for a command. The key will be the RESTORE key. ie:an NMI interrupt so whatever is running, I know that I can still get to issue commands to the operating system. Commands to the operating system such as LOAD will be needed. The following commands were considered necessary :-

LOAD - load a program from disk
RUN - Run a loaded program
EXEC - load and run a program from disk
SUSPEND - Suspend a running program
DISPLAY - display all programs currently loaded and their status (running or suspended).
QUIESCE - Stop and remove a program from memory.

The suspend state is when a program is actually in memory, but the CPU never actually processes any of the logic.

From this I have now described the basic elements of a simple multitasking operating system. There are of course other components which are not as critical to the system but are necessary and provide useful function. From here I will describe the components of the system, providing the assembler code which is commented Included also will be further descriptions and diagrams of the design and problems overcome in the design.
The code was written using the Lazy Grennius Assembler package available from ICPUG public domain software.

The modules making up the package are as follows :-

EQUATE - general labels and page zero equate map
INIT - initialise the MultiTasking operating system
COMMON - Common useful routines
TAB - maintain the program table
STOR - maintain the storage table
IRQ - control IRQ interrupts (program swoppong)
NMI - process NMI interrupts (command entry)
LOAD - load a program into memory
DISP - process the display command
RUN - process the RUN command
RELOC - relocate a program in memory
PROGSTOP - process SUSPEND and QUIESCE commands
MESS - Output messages to the screen
SCREEN - Screen control for the operating system
CLEAR - Clear the screen

The calling structure of these modules being as in Fig. 6 opposite.
The top box being the main routine. Any box stemming off from another means that it calls that routine.

STORAGE TABLE ROUTINES

The operating system has to know what memory is available for use and what memory has been used. This means some sort of table needs to be maintained with routines that can be called to perform functions on this table. The functions that are needed are

1. Locate a free block of storage based on a certain size and remove it from the table (thus indicating that memory isn't free)
2. Given an address of a memory block and it's length, add it to the table so as to indicate that this bit of memory can be reused (free storage).

Storage flag	Ram block	Start	End
1 byte	1 byte	2 bytes	2 bytes

The storage flag indicates whether the table entry is used or not.
1 = used. 0 = table entry free
The other areas define the available RAM.

RAM block takes on 2 values. 0 - RAM block 0 (equivalent to RAM configuration 3E). 1 - RAM value 1 (equivalent to RAM configuration 7E)

The table is of fixed size with fixed table entries. We need the flag to indicate whether a table entry is used or not.

There are 3 components to the storage module.
SETSTOR - initialise the storage table
LOCSTOR - locate a certain amount of storage and remove from the table
FRESTOR - Free up a block of storage. ie add to the table

SETSTOR

This routine will set up the storage table with the following information.

RAM 0 $3000 - $CFFF free
RAM 1 $1000 - $CFEE free
RAM 1 $E000 - $EEFF free

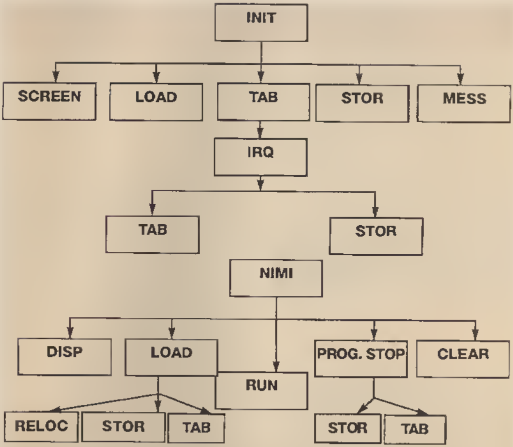
The area from $D000-$DEEF In both RAM 0 and RAM 1 is reserved for the I/O and the area from $1000 - $2EEE in RAM 0 is used for the MT operating system. The area $E000 - $EFFF in RAM 0 is used for the loader and will be explained later.

LOCSTOR

This routine will find an area of free storage via the table, and remove it from the table. The required input is :-

1. The size of the RAM block required

2. Option selection on where this RAM should come from
a) Any RAM block



THE ABOVE ROUTINES ALSO USE 'MESS','SCREEN'AND 'COMMON' AS WELL

b) A specific RAM block
c) whether the storage must start on a page boundary (page 0 and page 1 location)

FRESTOR

This routine does the opposite of the above It will add storage the to table to indicate that it can once more be used. If the table is full, a clean up process is initiated to try lo free up areas In lhe table. If it cannol, then an error message is produced

Input

Ram block of lhe storage
Start address of the storage
End address of the slorage

Output

Flag to say whether operating was successful

An extra piece of code within the free storage roulIne tries to clear up the table if il cannot find a free entry in the table. It scans the table trom lop to bottom concatenaling table entries where possible. It places concalenated informalion at the bottom end of the table rather than the top so as to allow further checking on this new information as it goes down the lable.

THAT CONCLUDES THE OPENING ARTICLE ON THIS SUBJECT. NEXT MONTH THERE WILL BE A PROGRAM EOR VIEWING AND PRINTING THE ASSEMBLER EILES THAT ARE ON THIS ISSUES DISK. UNTIL NEXT TIME THEN...READ AND INWARDLY DIGEST THIS MONTHS OFFERING.

HACKMAN AND SOBBIN'

The tale of an ex-computer widow. A harrowing story - definitely not for the squeemish or faint hearted

COMMANDER DOGGY

There was once a time, many, many aeons ago, when usage of the word 'hash' in our sheltered domain, conjured intense tremors of excitement, and instilled within our beating hearts, rare and multi-hued images of wild, free spirited musical jams.

For our music, our love, was the only universal language. To us, it was 'the sweet nectar of the spheres', 'the speech of angels' no less, and as such, there was no doubt whatsoever in our fevered, spaced out brains, that it could, with supreme ease, encompass the entire universe within it's raptuous, and transcending waves of purest, celestial energy!

THE STORY BEGINS

So what happened? I hear you all utter. Why did this fateful miracle not occur? A computer happened, a Commodore 64, to be precise, that's what!

Gone now were the days of creative genius, and instead, to lay claim to our modest living space, and to steal my spouses' (el Bones) undivided attention, and consequently, all of his time, was this solid state distractor, this mega intruder extraordinarius. Complete with it's rectangle power pack, endless infuriating games, and it's numerous incomprehensible tutors. All of which, I hasten to add, declared most adamantly..... 'Easy to understand languages', ie.... Basic, Machine Code, and would you believe it, 'C'..... !* (I thought that was a big blue wobbly thing that you swam in).

"Now just hold on there one goddamn' minute," I hear all you computer freaks exclaim with indignation, "what's so terrible about that?"

"Well please let me tell you what's so bad about that."

THE PLOT THICKENS

For a computer to be installed within a VERY LARGE room. well, that's ALMOST ACCEPTABLE But for a techno-creature to be housed within quarters the size of a SMALL SHOE-BOX, and positioned precariously upon the edge of a dilapidated coffee table, which lies only a foot or so in front of the comfy 'ole settee, well, that's QUITE A DIEEERENT MATTER entirely.

And especially when the said electro-gadget has no fine monitor to call it's own, and takes it upon itself to 'permanently borrow' the good old dream machine thats sits in the corner of the room, and usually mind warps it's viewers into regular nightly oblivion by erasing all their worries and woes. Then there's bound to be TROUBLE, and I mean BIG TROUBLE.

For to say that the 'updated abacus' eventually took over all our lives, is, with all honesty, no word of a lie. Friends that regularly came to visit our humble abode, especially to 'heed the weed', and to saturate their ears, and indeed their very souls with the fine sounds of our music, now would spend literally hours waiting impatiently for their turn to alight upon the increasingly sagging settee, in a wild attempt to do their bit for the poor and ailing games industry. Ho. Ho..Ho.. please excuse my stifled laughter.

IT GETS WORSE YOU KNOW

Games such as 'Munchie Men', '1985', 'International Soccer'. 'Dark Star', 'Exploding Fist', 'Frogger', 'Mercenary', to name but a handful, and not forgetting, of course, that marathon evoking 'piece de resistance', 'Elite', never ceased to be in motion. Our poor television set, forever possessed by this 'alien entertainment force', seemed destined never again to transmit it's usual array of goodies. My goodness, I know what you're thinking, how on earth did that poor soul survive without the steamy and erotic adventures of 'Sooty and Sweep', the sheer thrill of the scintillating 'Winner Takes All', or the earth and mind shattering 'Des O'Conner Show'. Well fact is, I almost did'nt!

Oh dear! oh dear! I suppose I really must come clean, otherwise I'll spend my entire life looking over my shoulder for fear of being prodded by an all accusing digit.

Okay, okay, don't twist my arm, ouch, ouch.....oooh, I admit it, I did hold a very sneaking admiration for the complicity and aesthetics of one or two of the games. Yes, yes, alright, so there was a certain something about them. And yes, okay, there was an innocent kind of childish magic that appeals to a certain level of mentality. Ouch.......stop that... oooooh, that really hurts... ... Help!

Yes, yes, you're right. I admit it, you win. I was the overall, mega champion of International Soccer for a whole eight consecutive weeks, and yes, yes, I absolutely adored it!

There you are   I hope you're satisfied. The lurid truth is finally out into the open. Now all that remains for me to do, is spill the remainder of the beens... (Oh no, not THE beans!!)

THE TRUTH IS OUT

Here goes then....... .. I Jenni Simpson, aged (mind your own business), was THE COMMANDFR DOGGY, Rugged Commander of the sturdy Cobra Mk II. Galaxy Hopper Magnificentis Status Dangerous, without doubt, and Thargoid obliteration my speciality. As a member of the Elite crew, I wore the badge ot allegiance with honour, and swore that whatever the circumstances, I would not hesitate to 'explore strange new worlds, to seek out new life forms and new civilisations, and to boldly go where no one had gone before'!

Back now to the zombie gamesters... .. .

There they would be... Whoops! There WE would be, seated bodies rigid with concentration, eyes popping and straining towards the television screen, and our ears totally deaf to all sound, save for the repetitious bleeps, blips, burps, and other computerised melodia that watted forth from the speakers. All other activity immediately ceased. Time stood still, literally trozen in space, as dozens of 'new addicts' chanced their arms to save yet another of their tive lives, in order to escape the evil mutant, or to avoid the lethal blows of the Kung Fu Master. Days rolled into night. Night rolled into day. Whilst piles of dirty crockery grew high in the sink, and motes of dust settled as thickly as new born snow about us! And then one fine day in Spring, many, many weeks later, as rapidly as this mania had begun, interest began to wane. "Nother go Sandy?" "Nah, nah thanks!" "Shamy, wanna go?" "No thanks, I'm a bit bored with it, actually!"

Were my perceptions deceiving me, or was this major threat to our sanity and lifestyle about to be thrown straight out of the window   No, no such luck!

THE SERIOUS STUFF

Now in an effort to fill the vast, empty chasm that had just opened up within my partner's life, he took to trying to program the computer in the increasing hope that he could simulate a new and revolutionary type of engine, that he had, over the past few years been developing. Discovering however, the project to be far too ambitious, he began to practise diligently each day, in a concentrated attempt to master the language of 'Basic'   His reasoning behind all of this copious, mental activity, being that if he could become adequately proficient in the language, he would then be knowledgeable enough to simulate his 'beloved' engine

And so began a 'new phase' in our lives. No longer interested in playing music, computer games, or even debating the 'meaning of life' until all hours of the morning with our motley crew of stoned spaced cadets, John became serious and studious. Eventually, he insulated himself to such a degree, that none of our friends thought it even vaguely interesting enough to continue their visits. This was a miraculous and heart-warming occurence as far as we were concerned. For although our friends were very pleasant, we had, for many years, been continually bombarded with an incessant stream of visitors, and were growing increasingly tired of the situation. For as soon as one group left, yet another would be sure to turn up. They would arrive upon our doorstep literally in droves, and then, like vast, efficient armies of leaf-cutting ants, would trail through our humble home, consuming anything even vaguely edible in their wake, like a vast cloud ot voracious locusts.

You've heard of the expression, 'everything stops tor tea'? Well in our modest dwelling, the idiom 'tea never stops', would be more appropriate. For the amount of tea, coffee, milk, sugar, biscuits, and numerous other edible or drinkable commodities consumed over the years, by our insatiable and unending hoards, was enough to keep Messrs, Brooke Bond, Nescafe, Tate & Lyle, and McVities, in business for the next millenium.

THE ADDICTION GROWS

Computer addiction grew like a noose slowly tightening about our necks   And then one day to my horror, I returned home from an afternoon's shopping expedition, to discover that we no longer had a second bedroom! Had this 'space for dreams' been mysteriously spirited away to some far off land, perhaps even 'Weaveworld'?   No, not on your Aunt Nelly   You've guessed it, el Bono, whilst I'd been away, had turned our trusty little spare bedroom into a 'Puter Hideaway'. Gone now was our comfy 'Silent Night', and standing in it's place was the exceedingly uncomfortable 'Mega Byte'

From then onwards, things seemed to go from bad to worse   Almost every moment ot the day and night, John would steal away to his new found 'techno-room' in order to feed his growing addiction. Like a soul possessed, he would frantically tap away at his little set of keyboards (much like I'm doing now), until steam would rise from his reddened fingertips, and the sound ot the tapping would echo hollowly throughout the house, to relay morse code messages of the doom laden days to follow. By this time, of course, like a squirrel gathering nuts for the winter, he had accumulated a vast array of computer peripherals and accessories, ie....a portable colour t v (for use as a monitor), a disk drive, a printer, a cuddley toy, a fondue set and loads of other bits and bobs, plus a shelf full of books and magazines, that would put even the most avid of bookworms to veritable shame.

Short of dynamite, there seemed to be no way of stopping him, even for a moment. And even when he wasn't actually working within his 'sacred temple', his eyes would become suddenly glazed, and he would drift off once more to that all consuming world of 'programming madness'. 'Munchie' times were the worst though. The amount of meals that have been lovingly prepared, and then allowed to grow cold, I couldn't begin to remember.

"Dinner's almost ready," I'd shout with enthusiasm.

"I'm on my way," would come back the distracted and pained reply... the minutes would tick slowly by...

"It's getting cold, John," I'd shout again... more delay ....

"J u s t c o m i n g" .... and then some fifteen minutes or so later, he'd emerge from the midst of the crypt, sporting a sheepish grin on his flushed face, to announce with fervour, that he had just completed yet another subtle routine or section of crucial code!

"Brilliant," I'd reply, "but your Veggie Lasagne, has seen much better days!" And then one fateful day it happened. After many failed attempts of sending his painstakingly created work to the best, and most well known of all the computer magazines in the whole wide world (crawl, crawl, plug, plug), he was offered... Da...Da ..Da...Dah... A CONTRACT.

THE DAWN OF TRUTH

They, or should I say, the good 'ole editor', had actually liked it! An abundance of thrills and spills and mucho gleeeee there was, but pause or cause for celebration, there certainly wasn't. Straight up the stairs to the 'room of torture' he immediatly went, spurred by his dazzling success, and eager to concoct yet another 'barn-storming, print-worthy proggie'.

My hero, my genius, my poor computer zombie! What was I to do, where could I turn to, who could help me to cure this terrible disease! And then, to top it all, as if my pain and suffering wasn't enough, I had yet another language to listen to. Wide eyed, straight-jacket laced, and bursting with zeal, the poor demented fool, would babble on for hours and hours telling me of such things as 'Bytes, nybles, buses, bugs, handshaking, and trashcans'. I now knew that if I heard one more word about anything even vaguely connected with computers, I would go stark, staring bonkers.

I omitted to mention, that we did have two very good friends who still came to visit us on occasions. We'd known them for years, and they were exceptionally well versed in all of our eccentricities, as well as computers. One of the friends, dear old Mally, or magical Mally as we affectionately called him, visited us every couple of weeks or so.

Upon one such visit, after he and John had spent (only four hours this time), up in the crypt yakking incessantly about ....Sshh. .you know what! Mally condescended to chat to the poor old woman about her latest project. (No, seriously folks, Mally's a marvellous friend, and I love him). Anyway, to cut a long story short, I was in the midst of writing a book. I'd written and illustrated a couple of childrens' books in the past, but this time fancied my hand at presenting my autobiography.

"It's a lengthy business," I was explaining to Mally, "it takes such a long time! Firstly you have to prepare all of your notes, then you have to translate the notes into rough copy. Next upon the agenda is the correcting proceedure, and then finally, you have to type the completed book at least two, three or evenmore times. A very time consuming process, and perhaps then, you may still discover bugs .....Ooops, did I say bugs......take me out and shoot me immediately!"

Mally listened with great intent, and then gazed off into the void whilst fingering his beard. "Think I've got just the thing for you," he said, with an allknowing smile. "How do you fancy a computer?"

THE BATTLE IS LOST

"A COMPUTER!" I repeated the words like a retarded parrot, with slow deliberation, the meaning taking a moment or so to infiltrate my grey matter. "Oh no, no", I uttered, and immediately dropped upon my knees and proceeded in garbled prose, to pray to the powers that be, to preserve me from such a monstrous thought.

Two hours and thirty valium later, and after I had been led away to my bed, a gibbering, babbling, nervous wreck, Mally came into my sanctuary, to inform me of just how invaluable a computer with good word processing software, would be to me!

"Just think," he said, with great conviction, "all you'd have to do would be to type in your story, and the computer would do all of the rest. There would be no frustrating rubbing out or tippex mania at all. Alterations would be carried out with the minimal amount of fuss, and when you're finally satisfied with the finished article, hey presto, hit a key, and the printer would print out the entire manuscript for you. Nothing could be simpler." He conclued with obvious satisfaction.

Honestly, I really, really did try to erase those heavenly thoughts of 'manuscript simplicity' from entering my line of reasoning, I even swore upon C.D.U's continued publication. But the more I tried, the more I began to see exactly what a boon this computer, and it's word processing package would be to me!

Mally, upon seeing that he'd made a minor breakthrough, decided to press on with this suble art of persuation, like a computer salesman on 'big heat'.

"Oh, you'd love it Jenni, you really would," he persisted. "Once you've used one, you'd never want to be without it," he assured me, with a truthfulness that would have put even the Buddha himself to shame. My defences were crumbling rapidly. Mally leapt in for the kill.

"I've got a computer at work," he oozed, "which would be ideal for you. It's only four years old, and our firm no longer uses it. It's a lovely machine," he soothed, "double disk drives, super little anti-glare monitor, and it's all in immaculate condition, and really cheap too!"

What could I do, what could I say, I was slowly being seduced by this silver tongued Adonis, I became helpless, my heart was putty in his hands. The laziness and sloth within me, trembled with turbulent waves of excitement, at the mere thought of LESS EFFORT. I was fain open, wanton, could do nothing save open my mouth and say, "give it to me, Mally, I want it. When can I have it? I want it now!"

Mally slouched back in his seat, took a long draw from a short cigarette and inhaled deeply. "I could bring it next week," he said with a grin as sly as a fox who had just persuaded a chicken to go out for a night on the town, "I know you'll just love it!"

And that was that. I was smitten. All of the trauma, all of the pain, and all of the sheer frustration and exasperation seemed to disappear overnight, once I had lain my eyes upon my glorious little Apricot! I embarked upon a fevered and wild intrigue with it, teasing it at first with my wide-eyed innocence, and then delving deeply into it's innermost workings, until there was nothing left that I did not know about it. We TWO became ONE. Me with all of my intimate, secret thoughts and feelings, and little 'Squishy', as I affectionately began to call him, with his unerring capacity to obey my every command.

For I have a floating theory, that all computer enthusiasts, are secretly frustrated megalomaniacs. Their sleek, stream-lined monitors, are in fact their balank, individual, little worlds. And as such, these empty worlds, devoid of all other intrusion, are unique places. Kingdoms no less, in which we are our very own mistresses or masters. For we can survey our universes, anytime we so wish, losing ourselves entirely within their infinite depths, to create what we will, where we will and when we will. Queens and Kings of our own destiny are we, and as such, hold the keys (or in this case the keyboards), to all doors which we chance upon.

No, seriously folks, what I say is this, if you can't beat them. . .join them. ....it's much simpler in the end.

For look what my trusty little fruit machine 'Squishy' and I have done, we've created a vaguely amusing, I hope, little ditty for you all to read.

Now all I can hope for, is that the good and celestial archangel Eves, in all of his glory and profound wisdom, will deem it fit for publication, and send to me loadsandloads of money, and make me reeeely rich and famous. Tee Hee.

So thank you all for reading my ravings, and I do hope that I've helped to put a few minds at rest. For you know, there really is **LIFE AFTER COMPUTING.**

And finally, the moral of this story is.......... ...

If you're feeling a little fruity, and you haven't had a byte,......grab yourself an Apricot, and make your nights alright! Boom! Boom!

CALLING ALL ADVENTURERS!! STOP

Don't miss out on a golden opportunity to obtain another superb Graphic Adventure from that master author; TONY ROME, and at the same time stand a chance to win one of the following great prizes.

● ● ● ● ● ● ●

1st prize
An Amiga 500 computer

2nd prize
A Colour Monitor for your C64

3rd prize
A Colour Printer for your C64

4th prize
A selection of software from DIAMOND BYTES

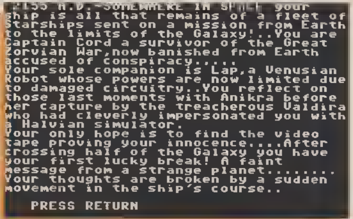
5th prize
5 winners will each receive a copy of the GAC+ Adventure Creation System from INCENTIVE SOFTWARE

● ● ● ● ● ● ●

As well as the prizes opposite, every single person that completes the Adventure will receive completely free another great **TONY ROME** game called **THE SPANISH TREASURE.**

Mr. **IAN ANDREW**, Managing Director of **INCENTIVE SOFTWARE, DIAMOND BYTES** and **TONY ROME** have got together to bring you this special offer.

The **ARGON FACTOR** is set in the year 2155 and places you in the middle of the universe with only a broken spaceship and an android for company. Your job is to repair your spaceship and clear your name. You will travel to different centuries in your quest and all your powers of logic and observation will be needed to succeed.



```
2155 A.D.-SOMEWHERE IN SPACE, your
ship is all that remains of a fleet of
Starships sent on a mission from Earth
to the limits of the Galaxy!..You are
Captain Cord a survivor of the Great
Zorvian War,now banished from Earth
accused of conspiracy.....
Your sole companion is Lap,a Venusian
Robot whose powers are now limited due
to damaged circuitry..You reflect on
those last moments with Anikra before
her capture by the treacherous Valdira
who had cleverly impersonated you with
a Halvian simulator.
Your only hope is to find the video
tape proving your innocence....After
crossing half of the Galaxy you have
your first lucky break! A faint
message from a strange planet........
Your thoughts are broken by a sudden
movement in the ship's course..

   PRESS RETURN
```

Please send me.......Copies of THE ARGON FACTOR at £9.95 each
Please send me.........Copies of CODEMASTER at £15.95 each
Please send me...........Copies of POWER TOOLS at £14.95 each
(These programs are available on disk only)

I enclose a cheque/postal order for . . . .
(Made payable to DIAMOND BYTES)

Name.. ....

Address ....

....

.... Postcode ....

Send to: DIAMOND BYTES 7 Graham Avenue Brinsworth
Rotherham, South Yorkshire
(Please allow 28 days for delivery)

The
ARGON FACTOR
will be available from 1st November 1990. Sole distributors are DIAMOND BYTES whose address you can find on the coupon opposite.

SUPER SNAPSHOT v5

Well, it may have taken a few years of hard work and five powerful versions, each one breaking new ground, but Super Snapshot has become the best cartridge in the world. The list below details the main features of Super Snapshot v5, if you need a little more persuasion look back to CDU issue 19, you'll be impressed.
So take a look at the red box you've got plugged in, and if our specifications knock it for six or you don't own a cartridge then don't just sit there, buy Super Snapshot v5 today!

FEATURES:

- ▶ All features available at the press of a button
- ▶ Works with all 64 (c) and 128 (D) computers
- ▶ Compatible with 1700/1764/1750 REU'S
- ▶ Snap any memory resident program into 1 file
- ▶ Save 7× faster and load 15× faster on the 1541, 1571 & 1581. Speeds of up to 25× faster when using Turbo 25 – even faster than Replay
- ▶ Super DOS Wedge
- ▶ GAME MASTER menu with sprite killer, infinitive lives generator and joystick port swapper
- ▶ Programmable function Keys
- ▶ Sprite Monitor
- ▶ Exclusive Character Set Monitor
- ▶ Exclusive Sound Sample Monitor
- ▶ Exclusive Boot sector support
- ▶ 300/1200/2400 Terminal program (40/80 column)
- ▶ SUPER DISK SNAPSHOT – our new super nibbler
- ▶ SCREEN-COPY now loads or saves in more formats and dumps in COLOUR to STAR LC10C printers and in 16 grey scales
- ▶ Improved full featured M/L monitor that DOES NOT CORRUPT MEMORY. Interrupt, examine and resume any running program
- ▶ Drive Mon
- ▶ BASIC PLUS with 15 new BASIC commands
- ▶ FILE MANAGEMENT SYSTEM – scratch, unscratch, rename or adjust skew. Includes our 1 or 2 drive file copier with partition support for the 1581
- ▶ Fast disk copier, 1 or 2 drives
- ▶ OUR FILE COPIER, DISK COPIERS and NIBBLER MAKE FULL USE OF THE REU'S
- ▶ Sequential file reader
- ▶ Utility disk
- ▶ Plus 150+ Kracker Jax parameters

ONLY £34.95

VISA ACCESS Prices include VAT & U.K delivery. Overseas orders send advertised price plus £2.50 for Airmail. Please send cheque, Postal Order or Credit Card details. VISA/ACCESS orders accepted by 'phone

**Masons Ryde,
Defford Road,
Pershore, Worcs.
WR10 1AZ.
Tel: (0386) 553153
Technical Support
Tel: (0386) 553222**

f/s/s/l COMPUTER SOFTWARE

ANIMATION STATION

Unleash your creativity with Animation Station, a powerful utility for generating all kinds of graphics on your Commodore 64 or 128. Built-in, predrawn pictures give you a head start on your creations. Automatic generation of circles, ovals, squares, boxes, straight lines, typography and other geometric shapes gives you all the tools of a self-contained, electronic drafting room. Combine type and graphics on the screen, draw in many colours, even connect your VCR to create titles and graphics for your home movies. Screen dump to your printer. Koala Compatible Package includes design touch pad, pencil and graphics software. Ideal for GEOS users.

ONLY £59.95

VIDEO BYTE 3

Digitize video images from your VCR, laser disk, B/W or colour camera, off the air or cable TV. New version 3.0 software features full re-display with multi-capture mode, menu select printing, expanded colourizing features, save to disk and much more. The hardware is no larger than an average cartridge which plugs into the User Port. The menu driven capture software is easy to use and pictures stored to disk can be imported into most popular drawing packages including GEOS. Prints in full colour to the Star LC10 when used with Super Snapshot.

ONLY £79.95

HOME VIDEO PRODUCER 64

Nothing can make home videos so special. Add titles, text, and even brilliant graphics to your favourite home videos with ease and the help of the Home Video Producer. You have the choice of 10 typefaces, 75 large full-colour graphics and ready made segments, but the most appealing aspect of the Home Video Producer is the ease with which you can do all this.

ONLY £29.95

C64 SLIMLINE UPGRADE CASE

Make your older C64 look like a newer model! All you need is a screwdriver and about 15 minutes to transfer your C64 insides to this new case. Complete instructions included.

ONLY £12.95

CARTRIDGE PORT EXTENDER CABLE

Are you cramped for space behind your computer? Is it hard to reach your cartridge port to plug-in or swap cartridges? This handy cable is the solution. The Cartridge Port Extender Cable connects to the cartridge port in the back of the computer and lets you plug in your cartridge to its other end. Since the cable is flexible, you can locate the cartridge up to 11" away for easier access. Not for REU's.

ONLY £19.95

GEOS APPLICATIONS

GEOSCAN ART

This special type of GEOS art has been created using The HandyScanner 64. Pictures are scanned at 400 dpi from magazines, books and papers to create the first geoSCAN ART Collection entitled The British Countryside and is packed full with Eagles, Owls and Butterflies.

ONLY £6.95

GEODIRECTORY

A comprehensive book listing all available GEOS programs. Details for each program is given including version numbers. The geoDirectory is divided into sections covering Paint, Write, Spell, File, Calc, Chart, Terminal, Graphic, Music, Animation, Games and many more.

ONLY £6.95

GEOTRONIX

A professional PCB designer utilizing the GEOS environment. Five double sided disks supply geoPublish with pre-designed components, sockets, edge connectors and layout grids in Photo Scraps. Using the Photo Manager and geoPublish the circuit is designed and printed.

ONLY £39.95

GEOS 64 V2.0	£29.95
GEOCALC 64	£24.95
GEOFILE 64	£24.95
GEOPROGRAMMER	£29.95
DeskPack Plus	£19.95
GEOCHART	£19.95
GEOS 128 V2.0	£39.95
GEOCALC 128	£29.95
GEOFILE 128	£29.95
GEOPUBLISH	£29.95
FontPack Plus	£19.95
Int Font Pack	£19.95

Commodore
DISK·USER·

Lineage 53p per word (+VAT)
Semi display £11 50 plus VAT per single column centimetre minimum 2cm.
Ring for information on series bookings/discounts
All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).
Make cheques payable to Alphavite Publications Ltd.
Send your requirements to
CLASSIFIED DEPARTMENT
ALPHAVITE PUBLICATIONS LTD., 20 POTTERS LANE, KILN FARM, MILTON KEYNES, MK11 3HF.

VISA

(0908) 569819

SOFTWARE

ONLY POOLS AND HORSES

FOOTBALL BOXFORM £14.95 cass £17.95 disk
Not a gimmicky plaything but a genuine statistical analysis of football form the sole objective of which is to improve the chances of winning on the pools or fixed odds. **Written by a pools expert the program has forecast over 50% more draws than would be expected by chance.** Homes, aways & draws are shown in order of merit and true odds given for every match. Merit tables show at a glance the teams currently in form and those having a lean spell. Separate AUSTRALIAN POOLS program included in the price.

RACING BOXFORM & HANDICAP WINNER
£14.95 cass £17.95 disk
Two programmes for the price of one. Boxform weighs up a race by considering the many facets of form. Easily amended to suit the user's ideas. HANDICAP WINNER shows the value for money bets. Over 1000 winners every year and ON DERBY DAY 1990 THE PROGRAM GAVE THE WINNERS OF 7 OUT OF 9 HANDICAPS. 20/1 10/1 7/1 11/2 4/1 4/1 6/4. **PROBABLY THE BEST DAY'S RESULTS EVER ACHIEVED, WITH OR WITHOUT A COMPUTER.**

SPECIAL OFFER: All the above for £24.95 (disks £29.95) plus a FREE program to work out almost any bet. So good it's used by bookies.

Why pay inflated prices? **BOXoft CLEVER** the same – get the **BEST.**
Advertised for five years in the sporting and computer press.
BOXOFT (CDU), 65 Allans Meadow, Neston, South Wirral, L64 9SO
Cheque/P.O./Access/Visa Tel: 051-336-2668 (24 hrs)

SOFTWARE

SILVER WING SOFTWARE
THE DEDICATED C64 P D LIBRARY
We have the best & latest quality demos utilities & games available
ONLY £2 00 on Tape or Disk
For the best benefits & power from your 64 you can not do without Public Domain Software
For our lists send an SAE to
SILVER WING SOFTWARE
185 CALLOWBROOK LANE,
RUBERY,
BIRMINGHAM
B45 9TG

SERVICES

COMMODORE SUPPLIES

C64 Power Unit	£21 99
C128 Power Unit	£39 99
C2N/C3N Cassette	£24 99
100 x 3½ Discs	£29.99
C64 Magnum Lightgun	£34 99
C64/128 Serial Cable	£5 99

Prices include VAT & P&P Chq/PO to
OMNIDALE SUPPLIES (CD),
23 Curzon Street, Derby, DE1 2E8
Tel 0332 291219

YOUR COMMODORE SPECIALS

TECH DRAW 64 – A comprehensive technical illustration aid for C64 owners for both disk and tape Commands available include LINE DRAW, RAYS, BOX, CIRCLE, ARC, ELLIPSE, FREEHAND, FILL, HATCH & TEXT Wipe copy, move and rotate, save and load or print your drawing all for £8.95 tape or disk Order Codes TD64T (tape), TD64D (disk)

SPEEDY ASSEMBLER – Your very own Assembler, a 100% memory resident program for loading from tape or disk, featured in the Your Commodore Machine Code Series and now the Y.C standard assembler only £9 95 Tape or Disk. Order Codes YSPASSD (disk), YSPASSR (tape)

ORDER FROM **CDU READERS' SERVICES,**
ALPHAVITE PUBLICATIONS LTD.,
20 POTTERS LANE, KILN FARM,
MILTON KEYNES, MK11 3HF.

Please make cheques payable to
Alphavite Publications Ltd.

Commodore
DISK·USER·

Lineage **53p** per word (+VAT)
Semi display **£11.50** (+ VAT) per single column centimetre Min size 2cm
Series discounts available on request All classified advertisements must be pre-paid
There are no reimbursements for cancellations
SEND TO CDU CLASSIFIED DEPARTMENT, ALPHAVITE PUBLICATIONS, 20 POTTERS LANE, KILN FARM, MILTON KEYNES MK11 3HF

I enclose my cheque/postal order for £ for insertions, made payable to Alphavite Publications.
(Delete as necessary)
PLEASE DEBIT MY ACCESS/VISA CARD NO: ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐
EXP. DATE.
£ FOR INSERTIONS
Name
Address Postcode
Daytime Tel No Signature Date

☐ FOR SALE ☐ SOFTWARE ☐ SPECIAL OFFERS ☐ REPAIRS ☐ HARDWARE ☐ DISKS

...it's dynamite!

POWER CARTRIDGE

FOR YOUR COMMODORE

64 / 128

SO MUCH FOR SO LITTLE

"...unbelievable value for money" ZZAP! Dec 89

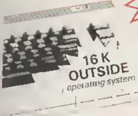
* POWER TOOLKIT
* POWER MONITOR
* TAPE & DISK TURBO
* PRINTERTOOL
* POWER RESET
* TOTAL BACKUP

..."Money well spent" YC/CDU Jan 90

42 Pg Manual - "Damned Good Handbook" CCI Jan 90

AVAILABLE FROM ALL GOOD COMPUTER RETAILERS

TRIED AND TESTED - OVER 100,000 SOLD IN EUROPE



..."highly recommended for C64 users" CCI Jan 90

YOU WILL WONDER HOW YOU EVER MANAGED WITHOUT IT

ONLY £16.99 INC VAT

A powerful BASIC Toolkit (Additional helpful commands) that considerably simplifies programming and debugging.

AUTO	HARDCAT	RENUMBER
AUDIO	HARDCOPY	REPEAT
COLOR	HEX$	SAFE
DEEK	INFO	TRACE
DELETE	KEY	UNNEW
DOKE	PAUSE	QUIT
DUMP	PLIST	MONITOR
FIND	PLOAD	PLOAD

RENUMBER	Also modifies all the GOTO's GOSUB's etc. Allows part of a program to be renumbered or displaced.
PSET	Set up of printer type
HARDCAT	Prints out Directory

The toolkit commands can be used in your programs.

Using POWER CARTRIDGE you can load up to 6 times faster from disk.
The Disk commands can be used in your own programs.

DLOAD	DVERIFY	DIR
DSAVE	MERGE	DEVICE
DISK		

MERGE	two BASIC programs can be merged into one.
DISK	With DISK you can send commands directly to your disk.

Using POWER CARTRIDGE you can work up to 10 times faster with your data recorder. The Tape commands can be used in your own programs.

LOAD	SAVE	VERIFY
MERGE	AUDIO	

A powerful machine language monitor that is readily available and leaves all of your Commodore memory available for programming.
Also works in BASIC ROM, KERNAL and I/O areas.

A ASSEMBLE	I INTERPRET	S SAVE
C COMPARE	J JUMP	T TRANSFER
D DIS-ASSEMBLE	L LOAD	V VERIFY
F FILL	M MEMORY	W WALK
G GO	P PRINT	X EXIT
H HUNT	R REGISTER	$ DIRECTORY
		DOS Commands

The POWER CARTRIDGE contains a very effective Printer Interface that self detects if a printer is connected to the Serial Bus or User Port.
It will print all Commodore characters on Epson and compatible printers.
The printer interface has a variety of set up possibilities. It can produce HARDCOPY of screens not only on Serial printers (MPS801, 802, 803 etc) but also on Centronics printers (EPSON, STAR, CITIZEN, PANASONIC, etc).
The HARDCOPY function automatically distinguishes between HIRES and LORES. Multi-colour graphics are converted into shades of grey. The PSET functions allow you to decide on Large/Small and Normal/Inverse printing.
The printer PSET functions are:

PSET 0	Self detection Serial/Centronics
PSET 1	EPSON mode only
PSET 2	SMITH-CORONA mode only
PSET 3	Turns the printing 90 degrees!!
PSET 4	HARDCOPY setting for MPS802/1526
PSET B	Bit image mode
PSET C	Setting Lower/Upper case and sending Control Codes
PSET T	All characters are printed in an unmodified state
PSET U	Runs a Serial printer and leaves the User port available
PSET Sx	Sets the Secondary address for HARDCOPY with Serial Bus
PSET LF	Adds a line feed, CHR$ (10), after every line
PSET LE	Switches PSET LF off

[illegible]

On the back of the POWER CARTRIDGE there is a Reset Button. Pressing this button makes a SPECIAL MENU appear on the screen.
This function will work with many programmes.

CONTINUE	Allows you to return to your program
BASIC	Return to BASIC
RESET	Normal RESET
TOTAL BACKUP DISK	Saves the contents of the memory onto a Disk. The program can be reloaded later with BLOAD followed by CONTINUE
RESET ALL	RESET of any program
TOTAL BACKUP TAPE	As BACKUP DISK but to TAPE
HARDCOPY	At any moment, prints out a Hardcopy of the screen. Using CONTINUE afterwards you can return to the program
MONITOR	Takes you into the Machine language Monitor

BDL

Bitcon Devices Ltd

**89 SEWICK ROAD
GATESHEAD
TYNE AND WEAR
NE9 1RS
ENGLAND**

**Tel 091 490 1975 and 490 1919 Fax 091 490 1918
To order: Access/Visa welcome Cheques or P/O payable to BDL
Price £16 99 incl, VAT,
UK orders add £1 20 post/pack total - £18.19 incl VAT
Europe orders add £2.50. Overseas add £3 50
Scandinavian Mail Order and Trade enquiries to Bhiab Elektronik, Box 216, Norrtalje 76123, SWEDEN Tel: + 46 176 18425 Fax: 176 18401
TRADE AND EXPORT ENQUIRIES WELCOME**

VIDEO COLOUR SPLITTER

£69.95 Inc Vat

AMIGA

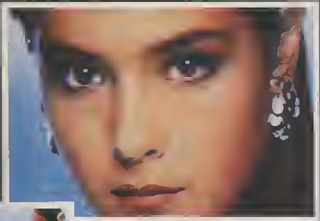
ST

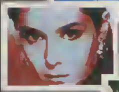
PC

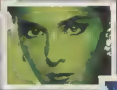
Actual Digitised colour screen - shots

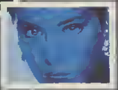
Vidi-RGB is an electronic filter which takes a colour video signal and separates it into the three primary colours (Red, Green and Blue) allowing each to be digitised.

Ideal for use with Vidi-Chrome & Frame Grabber or Digi-View Gold (By Newtek).









- For use with colour Digitisers replacing conventional Filter sets.
- Our Vidi - Chrome switches Vidi - RGB automatically grabbing full colour pictures in less than one second.
- Digitise full colour images direct from home VCR (must have perfect freeze frame)
- Digitise outstanding colour pictures direct from Canon's new Still Video Camera (an example shown on cover)
- Manual switching for maximum flexibility.
- Fully compatible with Digi - View Gold.

All these pictures are **actual unretouched screen-shots** illustrating the sequence of creating a full colour image using Vidi RGB, Vidi-Chrome and Vidi Amiga.

Limited

6 Fairbairn Road,
Kirkton North,
Livingston,
Scotland,
EH54 6TS.

Tel: 0506-414631
Fax: 0506-414634